The Influence of Cognitive Load on the Recollection of Information on Websites

Bachelor Thesis 2024

Yannis Bittner

Faculty of Behaviour, Management, and Social Sciences, University of Twente

202000383: Human Factors and Engineering Psychology (HFE)

First Supervisor: Dr. Rob van der Lubbe

Second Supervisor: Dr. Simone Borsci

January 24, 2024

Abstract

This research examines if the complexity of an item we are looking for on a website influences the recollection of information on that website. To examine this a change blindness experiment was performed. Twenty-four participants engaged in trials featuring three difficulty levels, in the form of requirements for a flight they should take while observing simulated flight booking websites. The difficulty levels serve as a cognitive load (CL) of the participants. The participants were presented with the flight booking website twice, separated by a mask, and were tasked with indicating whether they observed any changes. There were three types of trials possible, the first type was with no change occurring, the second type was with colour changes and the third type was with font changes. It was hypothesised that with increasing the difficulty level or CL the performance in the change blindness experiments decreases. Contrary to my hypotheses, while difficulty levels did not significantly impact memory recall, participants had a higher performance detecting font changes than detecting colour changes. Due to the absence of CL affecting the recollection of information on websites, it was assessed if participants experienced a learning effect by looking into their performances over time. Thereby a learning effect was found which may have contributed to the lack of hypothesised CL effects. This research showed some insights into CL and what variables and theories could have influenced the outcome of our research.

The Influence of Cognitive Load on the Recollection of Information on Websites

The internet plays a huge role in our daily lives from gathering information, streaming our favourite shows, staying in contact with people, purchasing goods, or booking vacations. A big part of the internet is the e-commerce sector, which reported an estimated 6.3 trillion US$ in sales revenue for 2023 and is estimated to represent 20% of the total sales worldwide (van Gelder, 2023). One of the earliest usages of the internet was the first reservation automation system “Sabre” which was built to keep track of offline ticket bookings by American Airlines (Boyd, 2023; Sabre, n.d., 2023). Nowadays, online flight booking has become widespread (Charlton, 2023). To have a successful website with a good user experience, website design plays a key role (Sudiana et al., 2021; Wei Shang & Ming Chun, 2010). One of the primary functions we use on any company website is the search function. In that process, our attention is focused on finding the item or service we need, which may lead us to overlook other items on the website. This study seeks to explore whether the complexity of the specific item we are searching for influences our memory of the presented information on websites. As part of this exploration, three distinct difficulty levels of instructions were employed in a change blindness experiment. To be able to link the impact on websites, the experiment uses a realistic approach where participants were tasked with observing simulated flight booking websites.

Visual Working Memory

Visual working memory (VWM) can be defined as the active retention of visual information for a brief period of time (Teng & Kravitz, 2019; Yatziv & Kessler, 2018). Its scope extends beyond solely preserving positional data, also including the retention of abstract features such as shape, colour, and texture (Ware, 2013). VWM is a newly used concept that is based on the older more general concept of working memory (WM), which also maintains and manipulates relevant information (Baddeley, 1996; Cowan, 1988). Thereby WM can hold a limited amount of information and can be applied during cognitive tasks (Cowan, 2014). The limited amount of information is also for the VWM a point of concern. For a long time, the limitation was assumed to be between seven plus minus two items (Miller, 1956) but a recent study found that this limit might be outdated and more items can be recollected (Suhani, 2023).

Human-Computer Interface

The human-computer interface (HCI) is a field in design that emerged in the 1980s when personal computers became more popular and using a computer no longer required programming skills (Carroll, 2014). It focuses on the graphical and interactable presentation of computer programs with usability and user experience in mind (Mandel, 2003; Waddell et al., 2016). A recent study by Ma, Wang, and Xue (2022) investigated the relationship between the aesthetics of interface elements layout and visual working memory. The study highlights the importance of considering cognitive abilities such as VWM while designing the graphical layout of an interface. In their study, they used aesthetics as a variable and showed that when people perceived websites as visually pleasing they improved the memory of the website. While our paper does not specifically emphasize aesthetics, the significance of the findings by Ma, Wang, and Xue (2022) cannot be overlooked. Their work underscores the importance of incorporating VWM considerations in website design, providing a foundation for future research to explore various aspects of VWM. To study VWM many studies choose to use a change blindness approach, described in the next section (He et al., 2022; Steinweg, 2021; Suhani, 2023).

Change Blindness

Change blindness (CB) refers to the phenomenon where humans struggle to notice significant changes in visual stimuli, even changes that one would think to be easily noticeable (Rensink et al., 1997; Simons & Rensink, 2005). Researchers have often investigated CB by showing subjects a blank page between viewing an object and the same object with one or more changes. The blank page serves as a disruption by overwriting iconic memory, contributing to the difficulty in detecting changes. In the past few decades, a lot of studies have been conducted exploring the scope of this phenomenon (He et al., 2022; Pashler, 1988; Phillips, 1974; Rensink et al., 1997; Simons, 1996; Simons & Rensink, 2005; Steinweg, 2021; Suhani, 2023).

One of the studies using a CB approach experiment was the study from Suhani (2023). He showed a fictional search machine website for two seconds next to a grey screen with a fixation cross as a masking screen for one second. Followed by either the same website or a slightly different version where the colour, shape or colour and shape of one of the items was changed. Suhani found that colour and, colour and shape changes were more noticeable than just shape changes. The author’s main focus was to determine the number of items someone can hold in their VWM.

Cognitive Load

The concept of cognitive load (CL) can be described as the amount of mental focus dedicated to a secondary task while trying to retain a memory task (Barrouillet et al., 2004, 2007; Orru & Longo, 2019). Recent studies have shown that a higher CL increases CB (Gunnell et al., 2019; McCarley & Vais, 2004). Based on CL the Time-Based Resource-Sharing (TBRS) model was created by Barrouillet et al. (2004). The TBRS model assumes that elements in WM are activated representations with decreasing activation levels due to time decay (Barrouillet et al., 2004; Glavan & Houpt, 2019; Puma et al., 2023). In this model, attention is regarded as the singular cognitive resource, capable of focusing on only one memory trace at any given time. To counteract the time decay of information activation, the attentional focus must be applied to each item, refreshing it and elevating its activation level. To be able to keep access to multiple items the items need to share their time of the focal point of attention. If an item does not get enough time, it will be forgotten.

For example, I study from Vergauwe et al. (2009) wanted to investigate the structure, functioning and recall performance of visuo-spatial WM. (Logie et al., 1991)Vergauwe et al. (2009)In their performed experiment, participants were tasked with remembering either a ball movement or a pattern with colour discrimination. After a 1500 ms interval dedicated to remembering their assigned item, participants engaged in a secondary task. For those remembering a pattern, the secondary task involved judging if another pattern had the same colour as the one, they were instructed to remember. Meanwhile, participants tasked with recalling a ball movement had to judge if a box fit into a designated space. The CL was manipulated by using three, five, or seven items in the secondary task. The results showed that participant performance declined with an increased CL suggesting that time influences recollection of the visuo-spatial WM. As predicted by the TBRS model, participant performance declined with an increased CL. Those findings are aligned with the predictions of the TBRS model.

Research Aim

This study seeks to explore VWM in the context of websites. Recognizing the integral role websites play in people's lives, we acknowledge that the process of searching for information on websites can impose a CL. This CL may influence the intake of information from the website, potentially leading to the oversight of relevant details crucial to the user's search. By investigating VWM in the context of websites, we aim to deepen the understanding of CL and investigate the potential impact of CL on users' ability to retain

relevant information during their online interactions. To be able to do so a CB experiment was chosen since it is a well-established method to explore both WM and VWM. Investigating the impact of CL is of particular interest due to its relatively recent emergence as a concept in cognitive research. The studies from Suhani (2023) and Vergauwe et al. (2009) were used as inspiration to combine a CB experiment with CL. Based on the TBRS model, we hypothesised that by increasing the difficulty level (or CL), the CB in participants would occur more often, i.e. their VWM would retain less due to greater CL. This thesis aimed to inspect the influence of CL on VWM by manipulating the complexity levels of instructions and recording participants' performance in a CB approach experiment. This resulted in the following research question: Has cognitive load an influence on the recollection of information on websites? The items of change in our experiment were colour and font changes. The colour change was chosen hance studies investigating CB commonly employ colour changes as a stimulus (Miežytė, 2019; Steinweg, 2021; Suhani, 2023; Varakin et al., 2007). Font change was chosen because of the lack of research within CB approach studies.

Methods

Participants

In total 24 participants took part in this study. They were recruited through convenience sampling by SONA, a system from the University of Twente to gather participants or they were asked by the researcher directly in person. The gender identity of the participants was 64% (15) female, 33% (8) male, and 4% (1) genderqueer, neither exclusively male nor female. Most participants were of European origin, except one from Australia, one from Costa Rica, and one from Venezuela. Eleven participants (46%) came from Germany, four (17%) from the Netherlands and nine (38%) from other countries. The age range was 19 to 39 with a mean of 23.15 (*SD* = 3,87). The majority of participants 67% (16) had at least a high school degree and 33% (8) had a bachelor's degree. One participant was noted to have a colour deficit with an outcome of 33% for their Ishihara colour blindness test (Ishihara, n.d.). The other participants had an outcome between 72 to 100%. All participants had normal vision acuity according to the FrACT10 Freiburg Vision test (Bach, 2006; Caltrider et al., 2023; DICOM Standards Committee, 2013).

Materials

The experiment was coded in Visual Studio Code 1.84.1 with Python 3.11.4 (Appendix A). Within the experiment two flight reservation websites were modelled after *kayak.com* (Figure 1 and 2). One version was with a blue colour theme and a flight from Amsterdam (AMS) to Rome. The other version had a green colour theme with a flight from Berlin (BER) to Lisbon. Both websites were created in Microsoft PowerPoint and designed to look almost the same except for the colours and different flights. The decision was made to construct two website versions to ensure diversity in display, aiming to sustain participants' interest and prevent disengagement. The experiment was executed on the laptop of the researcher a Lenovo IdeaPad Gaming 3 running Windows 11 with a 16-inch display and a resolution of 1920x1200. Most participants were seated with a viewing distance from the monitor between 40-80 cm, based on personal preferences.

The experiment utilised colour and font changes on the two flight reservation websites. The colour changes comprised changing the background of the logo, the top and left side banner, the coloured stripes for separation and all the coloured boxes with information inside like the date, destination, price and more (see Appendix B and Figure 1). The colour changes are task-relevant elements hence the colour change occurred within the boxes containing relevant information for the task participants were instructed to perform. There was no colour change in the symbols and other buttons. The colour was changed by changing the saturation and luminance leaving the hue at the same degree. The decision to maintain a constant hue level was based on findings from prior research, indicating that even the small alteration in hue was highly perceptible (Miežytė, 2019; Suhani, 2023). For the font changes, all written text except the logo was changed to prevent a section from popping out if it had a different font (Appendix C and Figure 2). Font changes were identified as task-relevant elements due to the task's heavy reliance on written information. Without reading the information on the flight page the participants would not be able to succeed in the task.

Figure 1

AMS Website Variation without Change but with Red Dotted Lines Indicating Where the Colour Changes Occurred

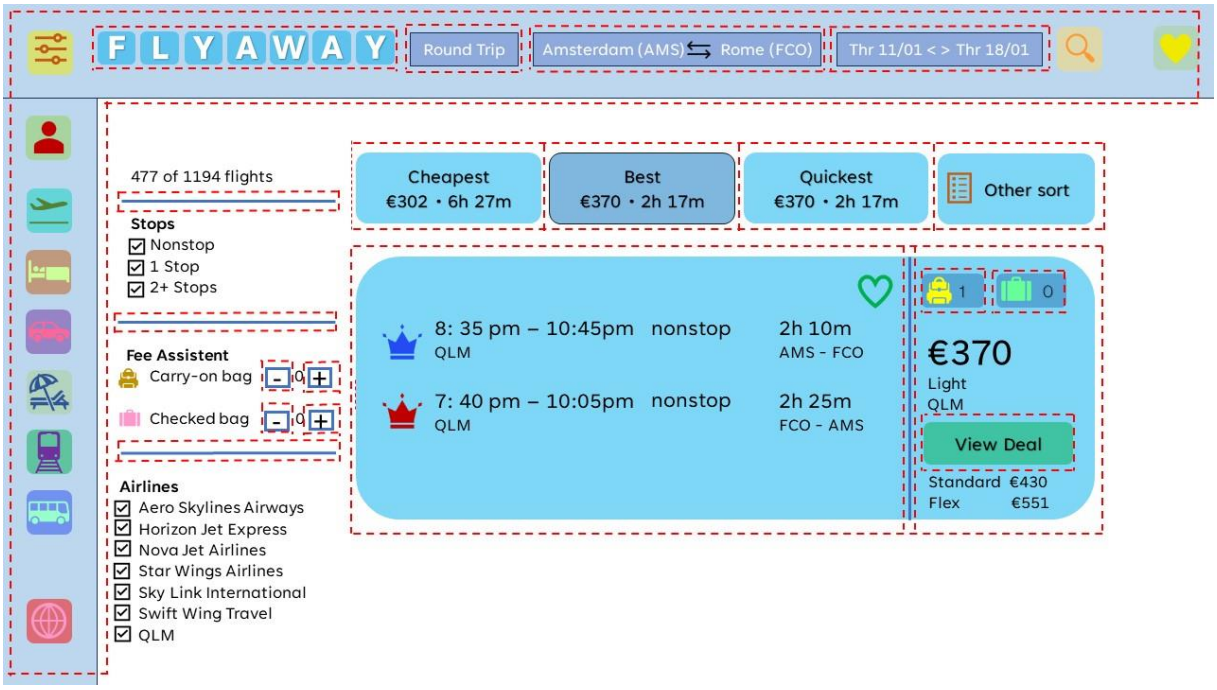
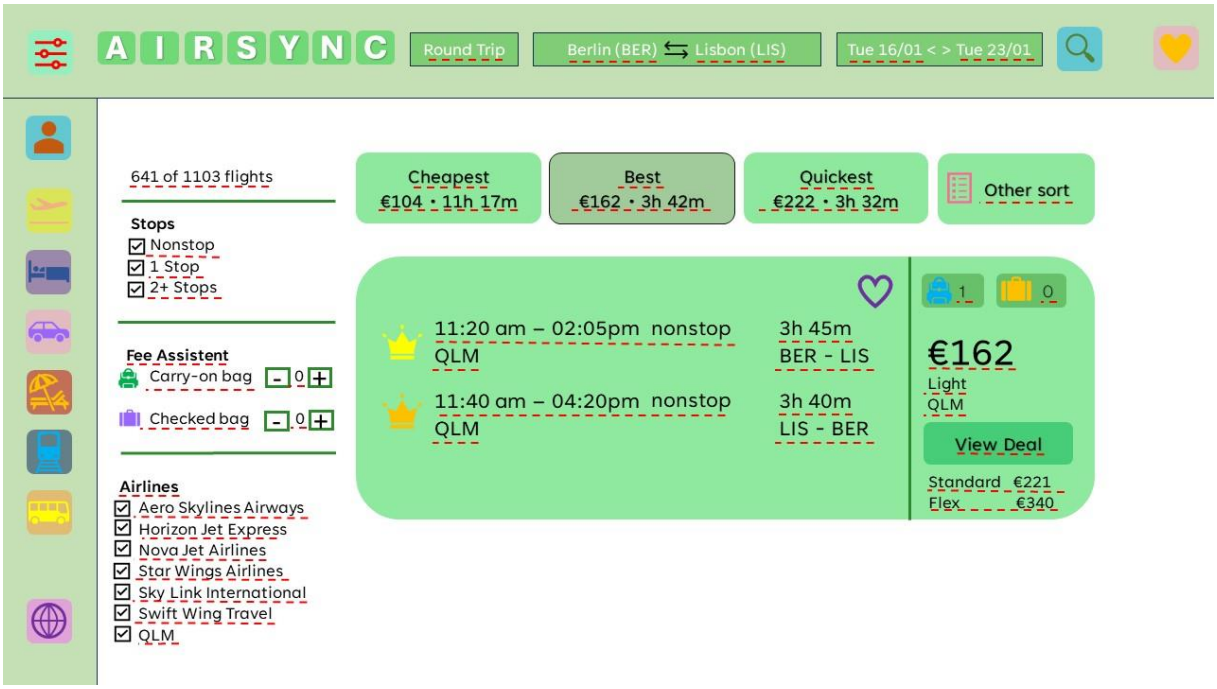


Figure 2

BER Website Variation without Change but with Red Dotted Lines Indicating Where the Font Changes Occurred



Procedure

The experiment was held with the researcher present in the room. Firstly, participants were asked to fill out the informed consent form which also explained the purpose of the experiment (Appendix D). Once consent was given, they were forwarded to the next section, the demographic questionnaire.

After participants finished both pre-tests, they were able to begin the experiment. The experiment was split into six blocks, each of which was followed by a two-minute break. The blocks were separated into three difficulty levels as well as two websites, resulting in six blocks. The difficulty levels consist of recalling one, two or three flight requirements like the date, time, price, airline etc. For example, "You are looking for a flight from Amsterdam to Rome" (difficulty level one, AMS), " You want to fly for a maximum of 150€ with an included checked bag for that price" (difficulty level two, BER) or "You are looking for a flight from Amsterdam to Rome from 11.01 to 18.01. Your flight budget is 400€" (difficulty level three, AMS) (Appendix E). The individual items to be recalled varied by website. Each participant received each difficulty level. There were twelve different orders of the blocks to counterbalance difficulty levels and website variation (Appendix F).

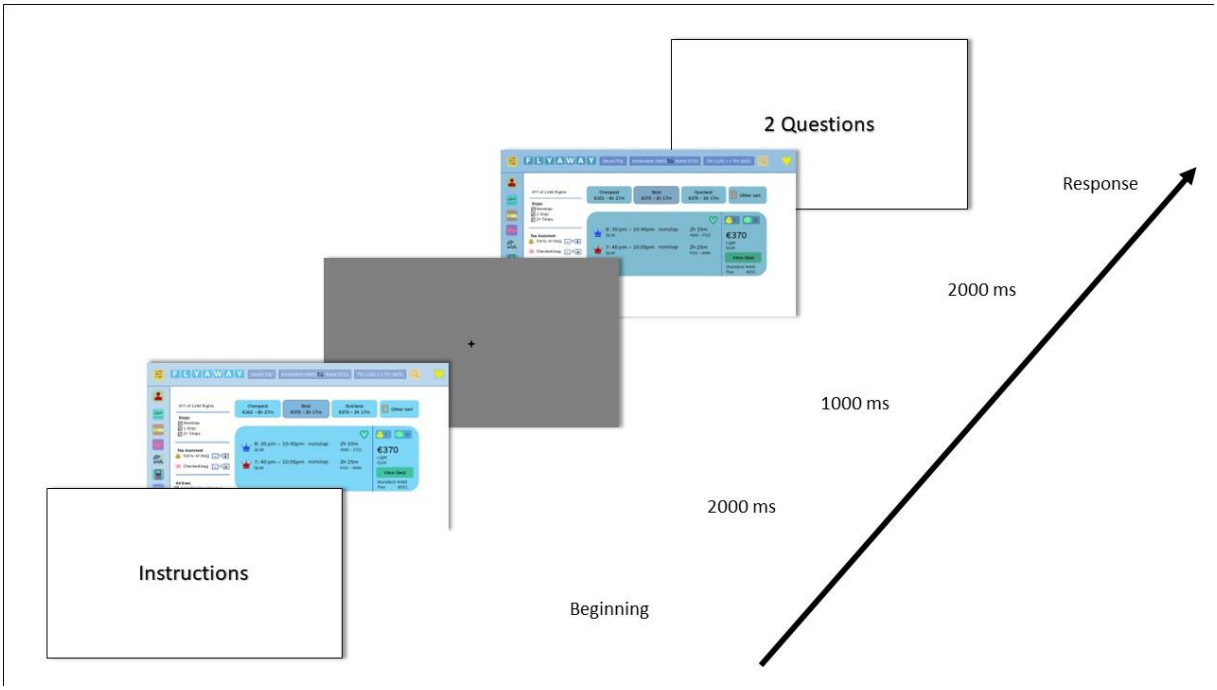
Each block consisted of 30 trials, a trial started with presenting instructed flight requirements in one of the difficulty levels. This was followed by the flight booking website, which was presented twice, for two seconds each, with a one-second mask in between (Figure 3). After the flight booking website appeared for the second time, participants were asked to answer the question "Does the offer fit your requirements for the flight?" followed by the question "Was there a change between the two websites?". Both questions could be answered with either "Yes" or "No". Each block type had six different flight requirements which got repeated five trials in a row creating in total six sub-blocks for each block. Half of the flight requirements matched the shown flight the other half did not match. The order of the sub-blocks was chosen randomly as well as which of the three different types of trials was presented. The three different types of trials were as follows:

1. The first type involved no change, with the first picture presented in the trial being identical to the second one.
2. The second type was a colour change, whereas the second picture included the colour change.
3. The third type was a font change, whereas the second picture included the font change.

No change occurred in 14 out of 30 trials in each block, the colour change occurred in eight out of 30 trials as well as the font change, resulted in 16 out of 30 trials with a change. Altogether, the experiment comprises six blocks, each consisting of 30 trials, resulting in a total of 180 trials. In the end, after the participants finished all the trials they were asked if they used any strategies during the experiment and if they wanted to leave some feedback.

Figure 3

The Setup of one Trial in the Order of Events



Data Analysis

Signal detection theory was used to calculate the d-prime (d') and beta (ß) for each participant for each difficulty level for both the colour and the font change. To be able to calculate the d' the hit rate and false alarm rate were computed. Subsequently, z-scores were calculated from these rates, followed by subtracting the false alarm rate's z-score from the hit rate's z-score. Some participants had a hit rate or false alarm rate of either 1 or 0 resulting in infinite d'. To prevent an infinite d' Stanislaw and Todorov's (1999) log-linear recommendation was applied. This recommendation dictates that if the hit or false alarm rate is a 1 or 0 it needs to be recalculated by increasing each hit/false alarm by an increment of 0.5, while the counts of signal trials and noise trials are augmented by 1.

The d' was calculated in this study to be able to indicate the performance of the participants (Keating, 2005). Keating suggests that the highest possible d' is 6.93 but the

highest level of effectiveness is represented by a d' value of 4.65. While the average d’ value generally hovers around 2, a lower d’ value of 1 corresponds to an approximate 69% correct response rate for both identical and different trials. Based on his findings and previous research it was settled that a d' value of 1.5 would be labelled as good, values ranging between 1 and 1.5 as average, and values below 1 as poor (Keating, 2005; Suhani, 2023).

The ß values were calculated to be able to estimate the response bias of each participant, i.e., how willing the participants were to say a change was present (Gardner et al., 1984). Gardner et al. (1984) showed that a ß value below 1 represents that the participant was liberal about reporting a change. A ß value above 1 represents that the participant was conservative in reporting a change.

After d' and ß were calculated a Shapiro and Wilk’s test was performed to check the normality and homogeneity assumption. With the calculated d' values, it was possible to determine whether the three different difficulty levels influenced the participant’s capacity to detect changes in the stimuli accurately. A t-test was performed to assess whether the performance of participants was affected by the differences between the two distinct flight pages, namely AMS and BER. Following this analysis, a decision was made regarding whether this factor should be included in subsequent analyses or omitted. Afterwards, a 3 difficulty levels (1, 2 and 3) x 2 changes (colour and font change) three-way mixed ANOVA analysis was conducted once for d' and once for ß. For the final step, to check for a learning effect, a new variable was introduced Try, indicating the first and second time a participant did a block. The d' and ß values for each participant and the changes in each block try were calculated. To investigate whether there is a difference in d' or ß between the first and second time a participant completed a block, a comprehensive 3 difficulty levels (1, 2 and 3) x 2 changes (colour and font change) x 2 tries (first and second try) three-way mixed ANOVA was carried out once for d' and once for ß. The three-way mixed ANOVA analysis for d' was done to check for indications of a learning effect between these blocks. All the data from the experiment were analysed with RStudio version 2023.09.1 and Excel version 2310 (Appendix G). The data from the questionnaire was analysed with the tools from Qualtrics and R with RStudio.

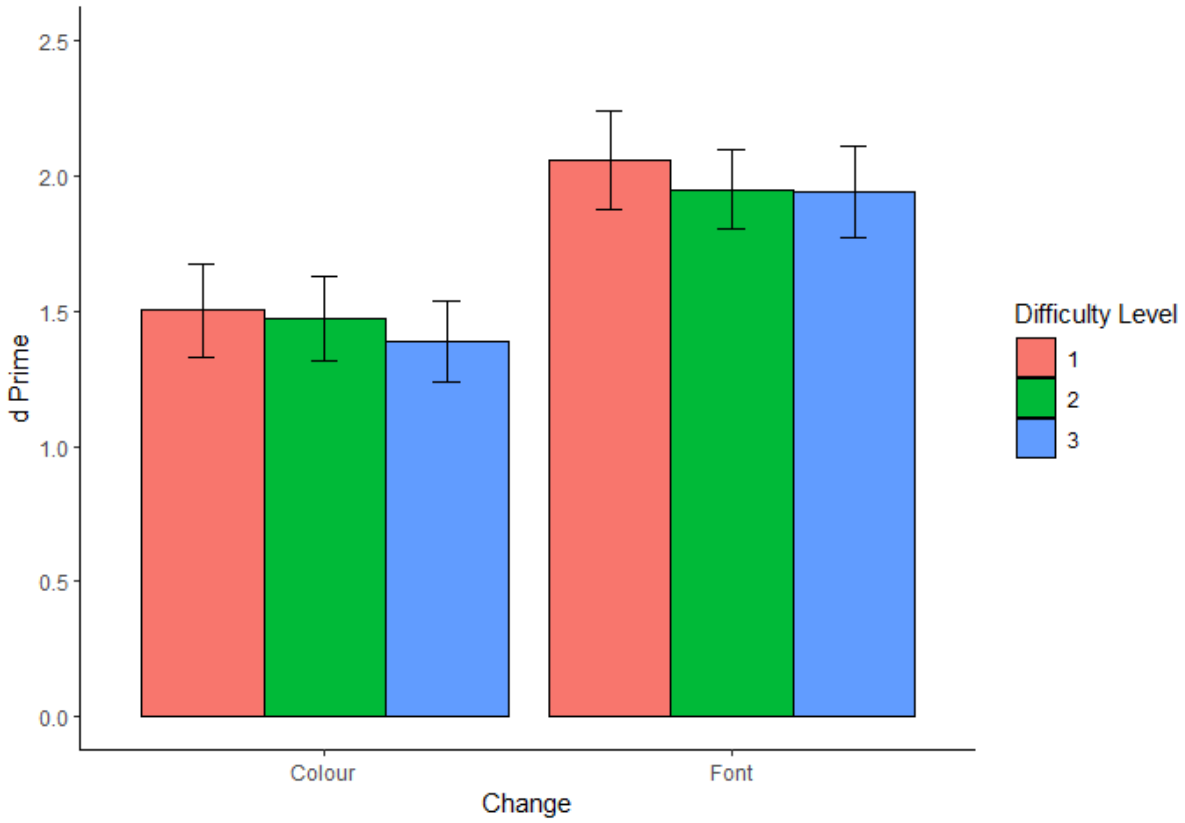
Results

Influence of Difficulty on Performance

This paper aimed to determine if CL in the form of Difficulty Level (1, 2, 3) influences the recollection of information on websites. To check this influence two website versions were created (AMS and BER) each with Change (colour, font). Figure 4 presents d' between Change and Difficulty Level. A small decrease in d' can be observed from difficulty level one to three. The figure also illustrates that across all three difficulty levels, the d' value increases by approximately 0.5 when transitioning from the colour change to the font change.

Figure 4

Bar Plot with Error Bars of d Prime between the Two Changes with the Three Difficulty Levels as Colours



A t-test between d' and the website variations demonstrated a non-significant main effect of websites on d', $t(141) = -0.7$, $p = .484$. Given the non-significant results and the fact that the primary focus of this paper does not centre on the distinction between the two website variations, it was determined to exclude this variable from further analysis. Next, a 3 (Difficulty Level) x 2 (Change) three-way mixed ANOVA analysis was executed to assess the

effect of Difficulty Level and Change on d' were performed (Table 1). There was a non-significant main effect of difficulty level, $F(2, 46) = 0.3$, $p = .729$. A significant effect of colour and font change on d' was observed, $F(1, 23) = 24.9$, $p < .001$. The interaction between the difficult levels and the changes was not statistically significant, $F(2, 46) = 0.1$, $p = .852$.

Table 1

3x2 and 3x2x2 Three-Way Mixed ANOVA Table with d-Prime

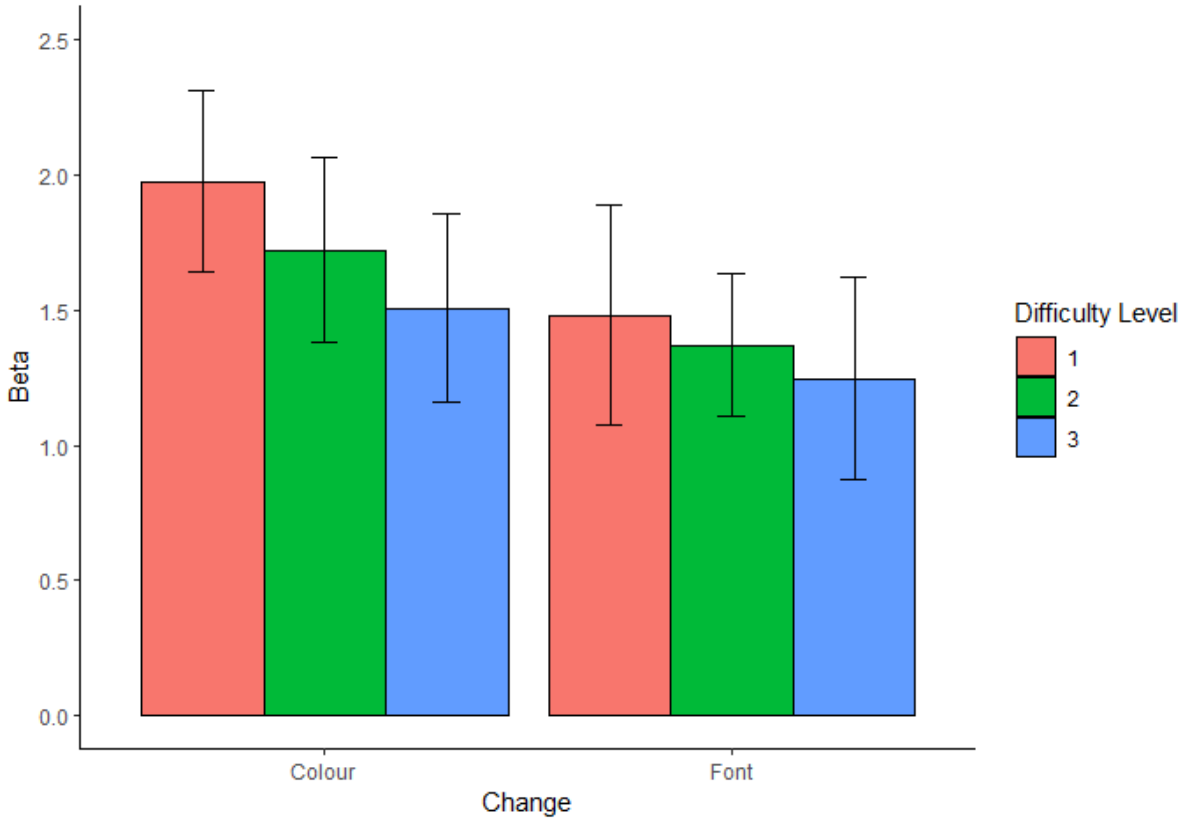
Effect	Degrees of Freedom	F-value	p-value
Difficulty Level	46	0.319	.729
Change	23	24.994	< .001
Difficulty Level-Change Interaction	46	0.161	.852
Difficulty Level	46	0.395	.676
Change	23	23.117	< .001
Try	23	13.402	.001
Difficulty Level - Change Interaction	46	0.008	.992
Difficulty Level - Try Interaction	46	0.54	.586
Changes - Try Interaction	23	9.06	.006
Difficulty Level - Change - Try Interaction	46	0.926	.403

Note. The results of the 3x2 three-way mixed ANOVA are above the separation in the table. The results of the 3x2x2 three-way mixed ANOVA for the learning effect are below the separation in the table.

Figure 5 presents the relationship between ß, Change and Difficulty Level. The ß values between Difficulty Levels in font change decreased from about 1.5 in difficulty level one to about 1.2 in difficulty level three. The ß value also decreased between Difficulty Levels in colour change from about 2 for the first difficulty level to about 1.5 for the third difficulty level.

Figure 5

Bar Plot with Error Bars of Beta between the Two Changes with the Three Difficulty Levels as Colours



To assess, Figure 5 presented relationships between ß, Change and Difficulty Level a 3 (Difficulty Level) x 2 (Change) three-way mixed ANOVA analysis on ß was performed (Table 2). There was a non-significant main effect of Difficulty Level, $F(2, 46) = 0.8$, $p = .439$ as well as of Change, $F(1, 23) = 2.8$, $p = .105$. The interaction between Difficulty Level and Change was also statistically insignificant, $F(2, 36.48) = 0.2$, $p = .719$.

Table 2

3x2 and 3x2x2 Three-Way Mixed ANOVA Table with Beta

Effect	Degrees of Freedom	F-value	p-value
Difficulty Level	46	0.837	.439
Change	23	2.845	.105
Difficulty Level-Change Interaction	36.48	0.263	.719
Difficulty Level	46	2.437	.099
Change	23	5.105	.034
Try	23	16.518	< .001
Difficulty Level - Change Interaction	46	0.339	.714
Difficulty Level - Try Interaction	30.87	0.275	.672
Changes - Try Interaction	23	1.147	.295
Difficulty Level - Change - Try Interaction	46	2.734	.076

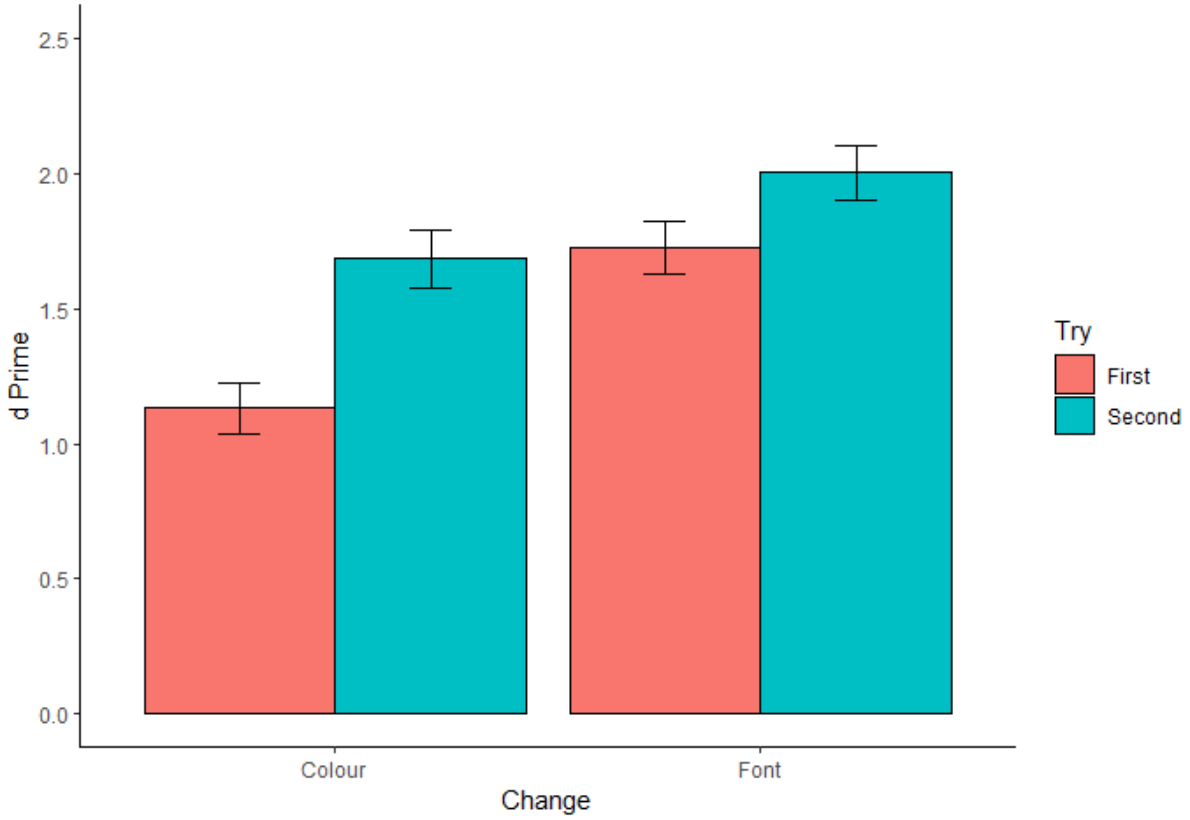
Note. The results of the 3x2 three-way mixed ANOVA are above the separation in the table. The results of the 3x2x2 three-way mixed ANOVA for the learning effect are below the separation in the table.

Learning Effect

Figure 6 presents an exploration of the significant interaction effect between the Change and Try. It can be seen that in both the colour and font changes the first try of a block had a lower d' than the second try. For the colour change the difference is about 0.6 and about 0.4 for the font change. Overall, for both tries font change had a higher ß than colour change.

Figure 6

Bar Plot with Error Bars of d Prime between the Two Changes with the Two Try's as Colours

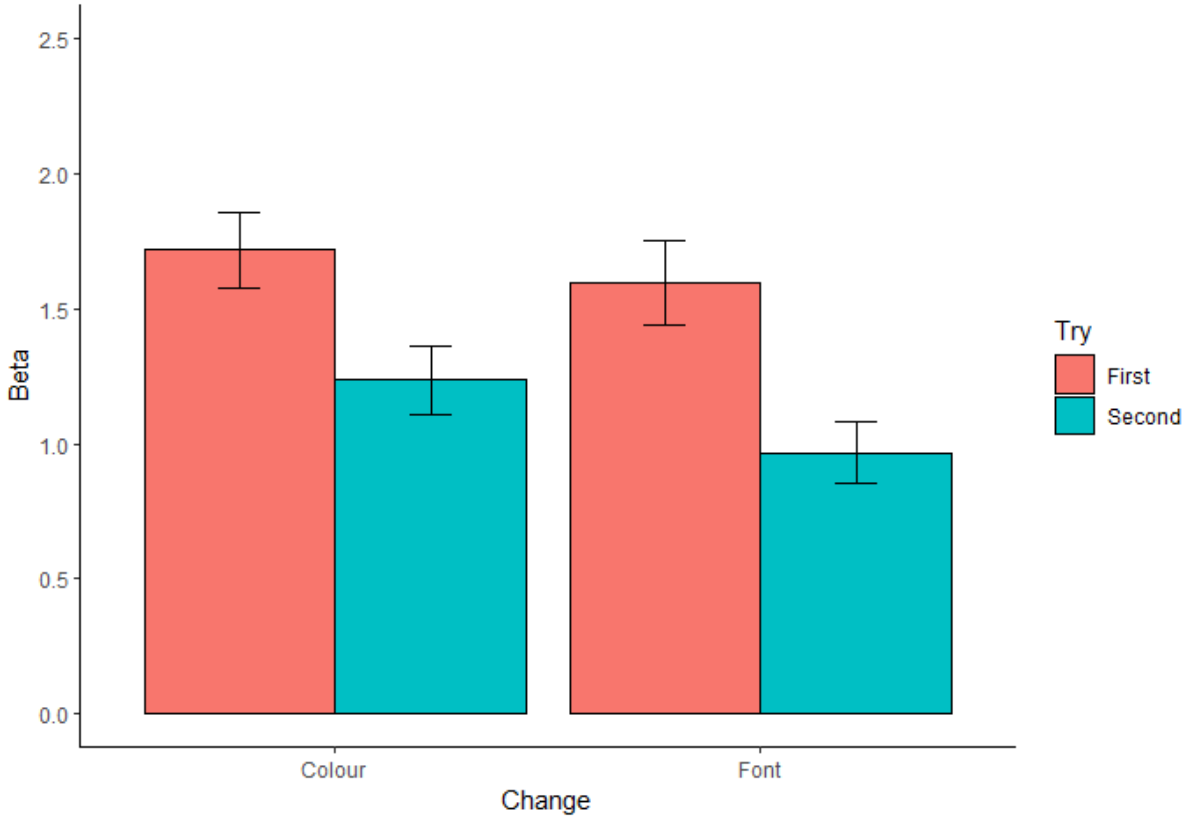


To assess for a learning effect a 3 (Difficulty Level) x 2 (Change) x 2 (Try) three-way mixed ANOVA was conducted (Table 1). Try showed a significant effect on the outcome, $F(1, 23) = 13.4$, $p < .001$. Interactions between Difficulty Level and Change, Difficulty Level and Try, as well as the three-way interaction (Difficulty Level: Change: Try), were all non-significant ($p > .05$). There was a significant interaction between Change and Try, $F(1, 23) = 9.06$, $p = .006$.

Figure 7 presents the relationship between ß, Change and Try. Both first tries of the colour and font change are about 0.2 apart whereas both second tries are about 0.3 apart. For both tries the colour change showed a higher ß compared to the font change. Between the first and second try of the colour change ß decreased by about 0.5, while the first try of the font change decreased by about 0.6 compared to the second try.

Figure 7

Bar Plot with Error Bars of Beta between the Two Changes with the Try's as Colours



To examine the relationship between ß, Change and Try a 3 (Difficulty Level) x 2 (Change) x 2 (Try) three-way ANOVA analysis was performed (Tabel 2). There was a significant main effect of Change, $F(1, 23) = 5.1$, $p = .034$ and of Try $F(1, 23) = 16.5$, $p = < .001$. A non-significant outcome ($p > .05$) was found in the interactions between Difficulty Level and Change, Difficulty Level and Try as well as the three-way interaction (Difficulty Level: Change: Try).

Discussion

This study aimed to investigate whether CL in the form of three difficulty levels affects the recollection of information on websites. In order to explore whether websites should take into account the complexity of the item's users seek, aiming to assist them in avoiding the oversight of crucial information. This study focused on deepening the understanding of CL. The effects of CL are usually measured in span task experiments (Barrouillet et al., 2007; Glavan & Houpt, 2019; Puma et al., 2023; Vergauwe et al., 2009). Based on span task experiments the TBRS model was created, it suggests that the gathered information of the WM needs attention and time thus the information does not deteriorate.

Therefore, it was hypothesised that with increasing CL, participant performance decreases. To examine this influence three difficulty levels were created and used in a CB experiment with colour and font changes on a flight booking website. Font changes were seen to be more task-relevant than colour changes since most of the information needed by the participants was written. The colour changes occurred partially in the background where important information was positioned therefore, they were seen as task-relevant as well but as less relevant. The outcome showed that most of d' results were good, only two values were average according to (Keating, 2005). The ß values decreased from conservative to less conservative between the difficulty levels and the first and second try (Gardner et al., 1984). The only time participants answered liberal at the second time they did the font changes. This decrease was only significant while looking at the first and second try of a block. The difference higher conservative answering in colour changes compared to the font changes was also significant.

The research question: “Has cognitive load an influence on the recollection of information on websites? ” was answered by examining the effect of specific experimental manipulations. The outcome of the analysis of d' showed that the performance of the participants did not statistically significantly differ based on our cognitive load manipulation. The unchanged performance across the different CLs is contrary to the main tenets of the TBRS model and the findings of Vergauwe et al. (2009). The TBRS model predicts that the impact of cognitive load on memory performance is expected to be consistent across various experiments when both memory set size and the ratio of cognitive load remain constant. Ricker and Vergauwe (2022) evaluated this by conducting four similar experiments where the memory set size and the ratio of cognitive load remained the same. In two of the four experiments, they found a cognitive load effect even though the TBRS model would predict similar outcomes for all four experiments. This suggested that the cognitive load effect has boundary conditions for occurring in the VWM. Based on their results they criticise the TBRS model, claiming it generalises the findings of cognitive load effects in verbal memoranda presented in complex span tasks to reflect universal characteristics of WM. However, their study focuses solely on complex span task experiments which makes it questionable whether their findings are transferable to CB approach experiments. Nonetheless, their findings of boundary conditions on the occurrence of cognitive load effect on VWM could be an explanation for why this study did not find an influence of CL on VWM.

The other outcome of our research was that the performance from font change was greater than the colour change, suggesting that font changes are easier to detect than colour changes. This could be because the letters moved during the font changes, or the colour

changes were more subtle, and therefore more difficult to detect than the font changes. Another explanation for that could be that font changes were more action-relevant elements than the colour changes which were also action-relevant elements but did not include important information for the flight requirements. The difference in task relance could therefore explain why font changes were more notable than colour changes. Due to the usage of the information, more attention was given to the text, Oberauer (2019) showed that attention plays a crucial role in what information is stored in WM. The theoretical framework "affordance competition hypothesis" from Cisek (2007) could also explain why font changes were more notable. In his hypothesis, he suggests that behaviour encompasses an ongoing competition between the opportunities currently available and the demands for action. In the case of our paper that would mean that the information in the written text, colours and symbols are all competing with each other to be further processed. Hence the written text held greater significance for the participants, suggesting that these pieces of information might have undergone more thorough processing, including details related to the font. This could explain why the font changes were more easily detectable but further research is needed to test if task relevance influences the recollection of information.

Due to the non-significant influence of CL on the VWM another ANOVA analysis was performed to check whether a learning effect could have influenced our results. Our analyses confirmed a learning effect between the first and second try. There was also a significant interaction found for Change and Try. This implies that the impact of change on d' varied depending on whether it was the first or second attempt. This learning effect could be another explanation for why there was no effect from the CL on the d' since the participants might have learned about the structure of the page, the instructions and/or where to look for the information leading to an eventual decrease in CL and increase in performance.

Limitations

While the discoveries regarding the influence of CL on the recollection of websites are valuable, it's crucial to recognise and confront specific limitations that likely impacted both the findings and the interpretation of the outcomes. Firstly, based on feedback from the participants, it was noted that several participants utilized the initial presentation of flight requirements to assess if the flight met the criteria, without checking for any changes. Multiple participants said that after they had determined that the flight data were not changing, they used the rest of the time to actively look for changes. That would mean that for four out of the five times, the requirements were shown, the participants did not experience a

CL. This would also explain why the three difficulty levels did not influence d'. It was tried to check if the first try of each instruction showed an influence from CL of the recollection on websites but due to the unexpected importance of this information, the data did not allow to extract reliable information from the first tries.

Secondly, the time showing the mask page between the two flight booking websites could have also influenced the performance of the recollection. Shapiro et al. (1997) reported that when two targets are presented shortly after each other, the recollection of the second target is not possible, coining this phenomenon of attentional blink. Usually, this happens when the two targets are presented less than 500 ms after each other, but a recent study has shown that this can differ from person to person Willems & Martens, 2016). In comparison, our research had a one-second break between the two flight booking websites which should be enough to prevent attentional blink, but in the studies about the attentional blink, relatively simple targets were used, and the participants did not deal with an additional CL. Because of the different frameworks, attentional blink may have influenced the results of this paper, by making it harder to perceive changes and potentially lowering d'. This could be the explanation that less important information was harder to recollect.

Thirdly, many participants indicated that they found it hard to detect the colour changes and easy to detect the font changes. This left the impression that the colour changes were too subtle and hard to see for many participants, especially since there were no hue changes in the colour making the colour changes less noticeable. That could explain the worse performance in the colour change. In future research, the noticeability of the colour change should be controlled for. In the font change, the spacing and the position of the letters differed making those changes more obvious.

Recommendations for Future Research

For future research, it can be recommended to explore why this study did not show a significant impact of CL on the recollection of websites. The research design could be changed to implement newly gained information in this paper. A potential design change could include focusing on the first try of each flight requirement. The number of repetitions of flight requirements could be reduced to increase the different requirements as well as to increase the time the participant can see the website the first time. That should allow the participants to find all the information on the website and to be able to look for changes. It should also be noted that the usage of new requirements for each trial could be itself a CL for the participants therefore it should be avoided. Additionally, a pre-test to define the best

colour changes for future participants could be conducted. That could ensure the outcome that font changes are more perceivable than colour changes if this will be found in future research.

Besides focusing on a different design approach to the experiment it could be also helpful to consider the impact of the attentional blink on the experiment. For further research changes in the experiment could be made to explore the influence of attentional blink by changing the time between the two websites among the participants. It could be explored if different times between the websites influence the performance of the participants. Another approach could also focus on the exploration of the boundary conditions of the VWM as described in Ricker and Vergauwe's (2022) papers and if it applies to CL in a CB approach experiment. To achieve that multiple experiments with different approaches could be conducted to see which approach shows an effect from CL on the VWM. Finally, it could be focused on what the effect of task-relevant elements is on the recollection of information. For that kind of research changes between task-irrelevant and relevant elements could be made and the performance analysed.

Conclusion

Concluding it can be said that this research contributes to our understanding of CL in VWM by indicating that CL might not significantly impact the recollection of information on websites. Additionally, it highlights the increased noticeability of font changes compared to colour changes. It also gave an important insight into what variables and theories could have influenced the outcome of our paper. To find out why CL did not influence the recollection of websites despite previous research suggesting it more research needs to be done. Consequently, future research should aim to either validate these findings through robust investigations or address the limitations encountered in this study that might lead to a different outcome.

References

Bach, M. (2006). The Freiburg Visual Acuity Test-Variability unchanged by post-hoc re-analysis. *Graefe's Archive for Clinical and Experimental Ophthalmology*, *245*(7), 965–971. https://doi.org/10.1007/s00417-006-0474-4

Baddeley, A. (1996). *Colloquium Paper This paper was presented at a colloquium entitled ' 'Memory: Recording Experience in Cells and Circuits* (Vol. 93). https://www.pnas.org

Barrouillet, P., Bernardin, S., & Camos, V. (2004). Time Constraints and Resource Sharing in Adults' Working Memory Spans. *Journal of Experimental Psychology: General*, *133*(1), 83–100. https://doi.org/10.1037/0096-3445.133.1.83

Barrouillet, P., Bernardin, S., Portrat, S., Vergauwe, E., & Camos, V. (2007). Time and Cognitive Load in Working Memory. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *33*, 570–585. https://doi.org/10.1037/0278-7393.33.3.570

Boyd, A. (2023). *C.R. Smith and SABRE*. University of Houston. https://engines.egr.uh.edu/episode/3117

Caltrider, D., Gupta, A., & Tripathy, K. (2023). *Evaluation of Visual Acuity*. StatPearls. https://www.ncbi.nlm.nih.gov/books/NBK564307/

Carroll, J. M. (2014, January 1). *Human Computer Interaction - brief intro*. Interaction Design Foundation - IxDF. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

Charlton, G. (2023, August 22). *Online Travel Trends & Booking Statistics + 12 Fascinating Stats 2023*. Sale Cycle. https://www.salecycle.com/blog/stats/digital-trends-travel-industry-12-fascinating-stats/

Cisek, P. (2007). Cortical mechanisms of action selection: The affordance competition hypothesis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*(1485), 1585–1599. https://doi.org/10.1098/rstb.2007.2054

Cowan, N. (1988). Evolving Conceptions of Memory Storage, Selective Attention, and Their Mutual Constraints Within the Human Information-Processing System. *Psychological Bulletin*, *104*(2), 163–191.

Cowan, N. (2014). Working Memory Underpins Cognitive Development, Learning, and Education. *Educational Psychology Review*, *26*(2), 197–223. https://doi.org/10.1007/s10648-013-9246-y

DICOM Standards Committee. (2013). *Reference Tables For Equivalent Visual Acuity Notations*. https://dicom.nema.org/dicom/2013/output/chtml/part17/sect_RR.2.html#:~:text=Normal%20visual%20acuity%20(which%20is,represented%20by%20negative%20logMAR%20values.

Gardner, R. M., Dalsing, S., Reyes, B., & Brake, S. (1984). Table of criterion values (ß) in signal detection theory. *Behavior Research Methods, Instruments, & Computers*, *16*(5), 425–436.

Glavan, J. J., & Houpt, J. W. (2019). An Integrated Working Memory Model for Time-Based Resource-Sharing. *Topics in Cognitive Science*, *11*(1), 261–276. https://doi.org/10.1111/tops.12407

Gunnell, D. O. A., Kunar, M. A., Norman, D. G., & Watson, D. G. (2019). The hazards of perception: evaluating a change blindness demonstration within a real-world driver education course. *Cognitive Research: Principles and Implications*, *4*(15). https://doi.org/10.1186/s41235-019-0165-4

He, C., Gunalp, P., Meyerhoff, H. S., Rathbun, Z., Stieff, M., Franconeri, S. L., & Hegarty, M. (2022). Visual working memory for connected 3D objects: effects of stimulus complexity, dimensionality and connectivity. *Cognitive Research: Principles and Implications*, *7*(19). https://doi.org/10.1186/s41235-022-00367-9

Ishihara, S. (n.d.). *Ishihara Instructions*. Retrieved November 27, 2023, from https://web.stanford.edu/group/vista/wikiupload/0/0a/Ishihara.14.Plate.Instructions.pdf

Keating, P. (2005). *D-prime (signal detection) analysis*. Spring. http://phonetics.linguistics.ucla.edu/facilities/statistics/dprime.htm#:~:text=The%20highest%20possible%20d'%20(greatest,corresponds%20to%20a%20d'%20of%201.0

Logie, R. H., Denis, M., & Marchetti, C. (1991). Visuo-spatial working memory: Visual, spatial, or central executive? In *Mental Images in Human Cognition* (pp. 105–115).

Ma, C., Wang, H., & Xue Chengqi. (2022). Research on the Relationship Between the Aesthetics of Interface Elements Layout and Visual Working Memory. In M. Kurosu, S.

Yamamoto, H. Mori, M. M. Soares, E. Rosenzweig, A. Marcus, P.-L. P. Rau, D. Harris, & W.-C. Li (Eds.), *HCI International 2022 - Late Breaking Papers. Design, User Experience and Interaction* (Vol. 13516, pp. 597–607). Springer International Publishing. https://doi.org/10.1007/978-3-031-17615-9

Mandel, T. (2003). User/System Interface Design. In H. Bidgoli (Ed.), *Encyclopedia of Information Systems* (pp. 535–549). Elsevier. https://doi.org/https://doi.org/10.1016/B0-12-227240-4/00190-8

McCarley, J. S., & Vais, M. J. (2004). Conversation Disrupts Change Detection in Complex Traffic Scenes. *Human Factors*, *46*(3), 424–436.

Miežytė, A. (2019). *Capacity and Structure of Visual Working Memory* [Bachelor Thesis]. University of Twente.

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, *63*(2), 81–97.

Oberauer, K. (2019). Working memory and attention - A conceptual analysis and review. *Journal of Cognition*, *2*(1), 1–23. https://doi.org/10.5334/joc.58

Orru, G., & Longo, L. (2019). The Evolution of Cognitive Load Theory and the Measurement of Its Intrinsic, Extraneous and Germane Loads: A Review. *Communications in Computer and Information Science*, *1012*, 23–48. https://doi.org/10.1007/978-3-030-14273-5_3

Pashler, H. (1988). Familiarity and visual change detection. *Perception & Psychophysics*, *44*(4), 369–378.

Phillips, W. A. (1974). On the distinction between sensory storage and short-term visual memory. *Perception & Psychophysics*, *16*(2), 283–290.

Puma, S., Matton, N., Paubel, P.-V., & Tricot, A. (2023). *Cognitive Load Theory and Time Considerations: Using the Time-Based Resource Sharing Model. Educational Psychology Re-view, inPress*. https://doi.org/10.1007/s10648-018-9438-6ï

Rensink, R. A., O'Regan, K., & Clark, J. J. (1997). The Need for Attention to Perceive Changes in Scenes. *Psychological Science*, *8*(5), 368–373. http://pathfinder.cbr.com/projects.

Ricker, T. J., & Vergauwe, E. (2022). Boundary conditions for observing cognitive load effects in visual working memory. *Memory and Cognition*, *50*(6), 1169–1185. https://doi.org/10.3758/s13421-022-01320-3

Sabre. (n.d.). *The Sabre Story*. Retrieved October 29, 2023, from https://www.sabre.com/files/Sabre-History-rev2017.pdf

Sabre. (2023). *About us*. Sabre.Com. https://www.sabre.com/about/

Shapiro, K. L., Arnell, K. M., & Raymond, J. E. (1997). The attentional blink. *Trends in Cognitive Sciences*, *1*(8), 291–296.

Simons, D. J. (1996). In Sight, out of Mind: When Object Representations Fail. *Psychological Science*, *7*(5), 301–305.

Simons, D. J., & Rensink, R. A. (2005). Change blindness: Past, present, and future. *Trends in Cognitive Sciences*, *9*(1), 16–20. https://doi.org/10.1016/j.tics.2004.11.006

Stanislaw, H., & Todorov, N. (1999). Article in Behavior Research Methods Instruments &amp Computers. *Behavior Research Methods Instruments & Computer*, *31*(1), 137–149. https://www.researchgate.net/publication/12804110

Steinweg, I. (2021). *Flexible Resources in Visual Working Memory for Color, Size, and Orientation* [Bachelor Thesis]. University of Twente.

Sudiana, Chandra, Y. U., & Angela, L. (2021). Key success factors for a better user experience in e-commerce website. *Proceedings of 2021 International Conference on Information Management and Technology, ICIMTech 2021*, 512–516. https://doi.org/10.1109/ICIMTech53080.2021.9535076

Suhani, A. (2023). *On the limits of short-term memory when exploring websites* [Bachelor Thesis]. University of Twente.

Teng, C., & Kravitz, D. J. (2019). Visual working memory directly alters perception. *Nature Human Behaviour*, *3*(8), 827–836. https://doi.org/10.1038/s41562-019-0640-4

van Gelder, K. (2023, September 26). *E-commerce worldwide - statistics & facts*. Statista. https://www-statista-com.ezproxy2.utwente.nl/topics/871/online-shopping/#topicOverview

Varakin, A., Levin, D. T., & Collins, K. M. (2007). Comparison and representation failures both cause real-world change blindness. *Perception*, *36*(5), 737–749. https://doi.org/10.1068/p5572

Vergauwe, E., Barrouillet, P., & Camos, V. (2009). Visual and Spatial Working Memory Are Not That Dissociated After All: A Time-Based Resource-Sharing Account. *Journal of Experimental Psychology: Learning Memory and Cognition*, *35*(4), 1012–1028. https://doi.org/10.1037/a0015859

Waddell, T. F., Zhang, B., & Shyam Sundar, S. (2016). Human-Computer Interaction. In C. R. Berger & M. E. Roloff (Eds.), *The International Encyclopedia of Interpersonal Communication* (First, pp. 1–9). John Wiley & Sons, Inc. https://doi.org/10.1002/9781118540190.wbeic0182

Ware, C. (2013). Visual Thinking Processes. In C. Ware (Ed.), *Information Visualization* (Third Edition, pp. 375–423). Morgan Kaufmann. https://doi.org/10.1016/b978-0-12-381464-7.00011-9

Wei Shang, F., & Ming Chun, T. (2010). Factors driving website success - the key role of internet customisation and the influence of website design quality and internet marketing strategy. *Total Quality Management and Business Excellence*, *21*(11), 1141–1159. https://doi.org/10.1080/14783363.2010.529335

Willems, C., & Martens, S. (2016). Time to see the bigger picture: Individual differences in the attentional blink. In *Psychonomic Bulletin and Review* (Vol. 23, Issue 5, pp. 1289–1299). Springer Science and Business Media, LLC. https://doi.org/10.3758/s13423-015-0977-2

Yatziv, T., & Kessler, Y. (2018). A two-level hierarchical framework of visual short-term memory. *Journal of Vision*, *18*(9), 1–26. https://doi.org/10.1167/18.9.2

Appendix A

Python code

In order to run the code, it is important to set the directory paths. The way I did it was that I had one folder with the flight websites. I had for each website variation one sub-folder. In those sub-folders, I had 30 more folders with each two pictures of the flight booking website. Eight folders had first the unchanged website picture and then the colour change and another eight had instead of the colour change picture the font change picture. 14 folders had two times the same picture inside. Then I had in a separate folder the pictures for the flight requirements. Within that folder, I had six sub-folders separated by difficulty level and website variation. Within each sub-folder, I had another layer of six folders including each one picture with the flight requirements. Half of the folders had matching flight requirements and half had not matching requirements.

```
import pygame
import os
import random
import csv
import time

# Pygame initialization
pygame.init()

# 12 specific orders of directory paths
block_order_1 = [0, 1, 0, 0, 1, 0]
block_order_2 = [1, 0, 1, 1, 0, 1]
sub_blocks_order_1 = [0, 3, 4, 0, 3, 4]
sub_blocks_order_2 = [1, 2, 5, 1, 2, 5]
sub_blocks_order_3 = [0, 5, 2, 0, 5, 2]
sub_blocks_order_4 = [1, 4, 3, 1, 4, 3]
sub_blocks_order_5 = [2, 1, 4, 2, 1, 4]
sub_blocks_order_6 = [3, 0, 5, 3, 0, 5]
sub_blocks_order_7 = [2, 5, 0, 2, 5, 0]
sub_blocks_order_8 = [3, 4, 1, 3, 4, 1]
sub_blocks_order_9 = [4, 1, 2, 4, 1, 2]
sub_blocks_order_10 = [5, 0, 3, 5, 0, 3]
sub_blocks_order_11 = [4, 3, 0, 4, 3, 0]
sub_blocks_order_12 = [5, 2, 1, 5, 2, 1]
# Change the variables
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
selected_order_sub_blocks = sub_blocks_order_1      # Old 2
participant_number = 0                              # Old 26
```

```
#
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Selecting the block order based on sub_blocks_order
if selected_order_sub_blocks in [sub_blocks_order_1, sub_blocks_order_3,
sub_blocks_order_5, sub_blocks_order_7, sub_blocks_order_9,
sub_blocks_order_11]:
    selected_order = block_order_1
elif selected_order_sub_blocks in [sub_blocks_order_2, sub_blocks_order_4,
sub_blocks_order_6, sub_blocks_order_8, sub_blocks_order_10,
sub_blocks_order_12]:
    selected_order = block_order_2

# Create and open the CSV file for writing
file_name = f"experiment_data_participant_{participant_number}.csv"
csv_file = open(file_name, "w", newline="")
csv_writer = csv.writer(csv_file)

file_name_T = f"experiment_time_participant_{participant_number}.csv"
csv_file_T = open(file_name_T, "w", newline="")
csv_writer_T = csv.writer(csv_file_T)

# Write the CSV header
csv_writer.writerow(["Participant_Number", "Block_Number",
"First_Picture_Filename", "Second_Picture_Filename", "Requirments",  "Change",
"Fond", "Colour", "Q1_Duration", "Q1", "Q2", "Q1_validate", "Q2_validate"])

csv_writer_T.writerow(["Participant_Number", "Block_Number", "Dif_Level",
"Duration_Sek", "Duration_Min"])

# Constants
WIDTH, HEIGHT = 1520, 875
GRAY = (128, 128, 128)
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
# Fonts with different sizes
LARGE_FONT = pygame.font.Font(None, 100)
MEDIUM_FONT = pygame.font.Font(None, 60)
SMALL_FONT = pygame.font.Font(None, 36)

# Calculation the centre of the screen
center_x = WIDTH // 2
center_y = HEIGHT // 2

# Initialize screen
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Yannis Bachelor Thesis experiment")
```

```
# Define button positions for centering by X coordinates with space in between
button_yes = pygame.Rect(WIDTH // 2 - 250, HEIGHT // 2 - 25, 200, 50)
button_no = pygame.Rect(WIDTH // 2 + 50, HEIGHT // 2 - 25, 200, 50)
next_button = pygame.Rect(WIDTH // 2 - 50, HEIGHT // 2 + 75, 100, 50)

# Function to display images
def display_image(image_path):
    image = pygame.image.load(image_path)
    image = pygame.transform.scale(image, (WIDTH, HEIGHT))
    screen.blit(image, (0, 0))
    pygame.display.flip()

# Function to handle questions
def draw_text(text, font, color, x, y):
    text_surface = font.render(text, True, color)
    text_rect = text_surface.get_rect()
    text_rect.center = (x, y)
    screen.blit(text_surface, text_rect)


# Welcome
def draw_welcome_screen():
    welcome_text = LARGE_FONT.render("Welcome to my Experiment", True, BLACK)
    welcome_rect = welcome_text.get_rect()
    welcome_rect.center = (center_x, center_y - 50)

    thanks_text = MEDIUM_FONT.render("Thank you for your participation!",
True, BLACK)
    thanks_rect = thanks_text.get_rect()
    thanks_rect.center = (center_x, center_y + 30)

    continue_text = SMALL_FONT.render("Press space to continue", True, BLACK)
    continue_rect = continue_text.get_rect()
    continue_rect.center = (center_x, center_y + 90)

    screen.fill(WHITE)
    screen.blit(welcome_text, welcome_rect)
    screen.blit(thanks_text, thanks_rect)
    screen.blit(continue_text, continue_rect)
    pygame.display.flip()

# Introduction
def draw_introduction():
    intro_text = LARGE_FONT.render("General Instructions", True, BLACK)
    intro_rect = intro_text.get_rect()
    intro_rect.center = (center_x, center_y - 150)

    text_1 = MEDIUM_FONT.render("The experiments goes as follows:", True,
BLACK)
    text_1_rect = text_1.get_rect()
```

```
    text_1_rect.center = (center_x, center_y - 60)

    text_2 = MEDIUM_FONT.render("First you get specific instructions for five
rounds.", True, BLACK)
    text_2_rect = text_2.get_rect()
    text_2_rect.center = (center_x, center_y - 20)

    text_3 = MEDIUM_FONT.render("Then you get automatically shown two pictures
in a row.", True, BLACK)
    text_3_rect = text_3.get_rect()
    text_3_rect.center = (center_x, center_y + 20)

    text_4 = MEDIUM_FONT.render("Afterwards there are two questions and then
one round is over.", True, BLACK)
    text_4_rect = text_4.get_rect()
    text_4_rect.center = (center_x, center_y + 60)

    text_5 = MEDIUM_FONT.render("There will be multiple rounds per section.",
True, BLACK)
    text_5_rect = text_5.get_rect()
    text_5_rect.center = (center_x, center_y + 100)

    text_6 = MEDIUM_FONT.render("After one section there will be a short
break.", True, BLACK)
    text_6_rect = text_6.get_rect()
    text_6_rect.center = (center_x, center_y + 140)

    text_7 = MEDIUM_FONT.render("In total there will be six sections.", True,
BLACK)
    text_7_rect = text_7.get_rect()
    text_7_rect.center = (center_x, center_y + 180)

    continue_text = SMALL_FONT.render("Press space to continue", True, BLACK)
    continue_rect = continue_text.get_rect()
    continue_rect.center = (center_x, center_y + 240)

    screen.fill(WHITE)
    screen.blit(intro_text, intro_rect)
    screen.blit(text_1, text_1_rect)
    screen.blit(text_2, text_2_rect)
    screen.blit(text_3, text_3_rect)
    screen.blit(text_4, text_4_rect)
    screen.blit(text_5, text_5_rect)
    screen.blit(text_6, text_6_rect)
    screen.blit(text_7, text_7_rect)
    screen.blit(continue_text, continue_rect)
    pygame.display.flip()

# Function to show a gray screen
def display_gray_screen(delay=0):
```

```
    screen.fill(GRAY)
    text_surface = LARGE_FONT.render("+", True, BLACK)
    text_rect = text_surface.get_rect()
    text_rect.center = (center_x, center_y)
    screen.blit(text_surface, text_rect)
    pygame.display.flip()
    pygame.time.delay(delay)

# Function to show buttons
def draw_buttons_1Q():
    draw_text("Does the offer fit your requirements for the flight?",
SMALL_FONT, BLACK, WIDTH // 2, 300)
    pygame.draw.rect(screen, WHITE if Q1yes == 0 else GREEN, button_yes)
    pygame.draw.rect(screen, WHITE if Q1no == 0 else GREEN, button_no)
    draw_text("Yes", SMALL_FONT, BLACK, WIDTH // 2 - 150, HEIGHT // 2)
    draw_text("No", SMALL_FONT, BLACK, WIDTH // 2 + 150, HEIGHT // 2)

def draw_next_button_1Q():
    if (Q1yes or Q1no):
        pygame.draw.rect(screen, WHITE, next_button)
    else:
        pygame.draw.rect(screen, GRAY, next_button)
    draw_text("Next", SMALL_FONT, BLACK, WIDTH // 2, HEIGHT // 2 + 100)

def draw_buttons_2Q():
    draw_text("Was there a change between the two websites?", SMALL_FONT,
BLACK, WIDTH // 2, 300)
    pygame.draw.rect(screen, WHITE if Q2yes == 0 else GREEN, button_yes)
    pygame.draw.rect(screen, WHITE if Q2no == 0 else GREEN, button_no)
    draw_text("Yes", SMALL_FONT, BLACK, WIDTH // 2 - 150, HEIGHT // 2)
    draw_text("No", SMALL_FONT, BLACK, WIDTH // 2 + 150, HEIGHT // 2)

def draw_next_button_2Q():
    if (Q2yes or Q2no):
        pygame.draw.rect(screen, WHITE, next_button)
    else:
        pygame.draw.rect(screen, GRAY, next_button)
    draw_text("Next", SMALL_FONT, BLACK, WIDTH // 2, HEIGHT // 2 + 100)

# Function to extract variables from the file names
def extract_variables_from_filename_1(filename):
    variables = {
        "Requirments": 0,
        "Dif_Level": 0
    }

    if "R" in filename:
        variables["Requirments"] = 1

    if "1" in filename:
```

```
        variables["Dif_Level"] = 1

    if "2" in filename:
        variables["Dif_Level"] = 2

    if "3" in filename:
        variables["Dif_Level"] = 3

    return variables

def extract_variables_from_filename_2(filename):
    variables = {
        "Change": 0,
        "Fond": 0,
        "Colour": 0
    }

    if "C" in filename:
        variables["Change"] = 1

    if "F" in filename:
        variables["Fond"] = 1

    if "K" in filename:
        variables["Colour"] = 1

    return variables

# End
def draw_end():
    end_1_text = LARGE_FONT.render("YOU MADE IT !!!!!!!!!", True, BLACK)
    end_1_rect = end_1_text.get_rect()
    end_1_rect.center = (center_x, center_y - 200)

    end_2_text = MEDIUM_FONT.render("Did you used any strategies doing the
experiment?", True, BLACK)
    end_2_rect = end_2_text.get_rect()
    end_2_rect.center = (center_x, center_y - 100)

    end_3_text = MEDIUM_FONT.render("If you like share some feedback about the
experiment.", True, BLACK)
    end_3_rect = end_3_text.get_rect()
    end_3_rect.center = (center_x, center_y + 0)

    end_4_text = LARGE_FONT.render("Thank YOU for your participation!", True,
BLACK)
    end_4_rect = end_4_text.get_rect()
    end_4_rect.center = (center_x, center_y + 100)

    screen.fill(WHITE)
```

```
    screen.blit(end_1_text, end_1_rect)
    screen.blit(end_2_text, end_2_rect)
    screen.blit(end_3_text, end_3_rect)
    screen.blit(end_4_text, end_4_rect)
    pygame.display.flip()

# Working directories
directory_paths = [
    r'C:\Users,
    r'C:\Users']

directory_paths_sub_blocks = [
    r'C:\Users\,     #0
    r'C:\Users\,  #1
    r'C:\Users\,     #2
    r'C:\Users\,  #3
    r'C:\Users\YVanc\,     #4
    r'C:\Users\YVanc\']  #5

# Shuffle the directory paths based on the selected order
shuffled_directory_paths = [directory_paths[i] for i in selected_order]
shuffled_directory_paths_sub_blocks = [directory_paths_sub_blocks[i] for i in
selected_order_sub_blocks]

# Used folders
running = True
used_folders_R_0 = set()
used_folders_R_1 = set()
used_folders_R_2 = set()
used_folders_R_3 = set()
used_folders_R_4 = set()
used_folders_R_5 = set()

used_folders_SB_0 = set()
used_folders_SB_1 = set()
used_folders_SB_2 = set()
used_folders_SB_3 = set()
used_folders_SB_4 = set()
used_folders_SB_5 = set()
used_folders_SB_6 = set()
used_folders_SB_7 = set()
used_folders_SB_8 = set()
used_folders_SB_9 = set()
used_folders_SB_10 = set()
used_folders_SB_11 = set()

# Initialize variables
Q1yes = 0
Q1no = 0
Q2yes = 0
```

```
Q2no = 0
Q1_variable_name = 0
Q2_variable_name = 0

# Pause
pause_duration = 120  # 2 minutes in seconds
end_duration = 10
start_time = 0

# Blocks
blocks = ["Block_0", "Block_1", "Block_2", "Block_3", "Block_4", "Block_5"]
sub_blocks =["Sub_Block_0", "Sub_Block_1", "Sub_Block_2", "Sub_Block_3",
"Sub_Block_4", "Sub_Block_5",
             "Sub_Block_6", "Sub_Block_7", "Sub_Block_8", "Sub_Block_9",
"Sub_Block_10", "Sub_Block_11"]

# Timer
start_time_Q1 = 0
time_taken_Q1 = 0
start_time_block = 0
block_durations = 0

# Initialize block index
block_index = 0
sub_block_folder_use_count = 0
last_round = 0

# Initial state
STATE = "Welcome"

# Main game loop
while running:
    current_time = pygame.time.get_ticks()
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
            if STATE == "Welcome":
                STATE = "Introduction"
            elif STATE == "Introduction":
                STATE = "Start_T"
            elif STATE == "Start_T":
                STATE = sub_blocks[block_index]
            elif STATE == sub_blocks[block_index]:
                STATE = blocks[block_index]
            elif STATE == blocks[block_index]:
                if (Q1yes or Q1no):
                    STATE = f"R_{block_index}_Question_2"
            elif STATE == f"R_{block_index}_Question_2":
                if (Q2yes or Q2no):
```

```
                    if not folders:
                        block_index += 1  # Move to the next block
                        if block_index < len(blocks):
                            STATE = sub_blocks[block_index]
                        else:
                            STATE = "End"
                            if STATE == "End":
                                STATE = "Quit"

    if STATE == "Welcome":
        draw_welcome_screen()


    if STATE == "Introduction":
        draw_introduction()

    if STATE == "Start_T":
        if not start_time_block:
            start_time_block = time.time()
            STATE = sub_blocks[block_index]

    if STATE in sub_blocks:
        directory_path_sub_block =
shuffled_directory_paths_sub_blocks[block_index]
        sub_block_folder_variable_name = f"used_folders_SB_{block_index}"

        if not globals()[sub_block_folder_variable_name]:
            globals()[sub_block_folder_variable_name] = set()

        sub_block_folders = [f for f in os.listdir(directory_path_sub_block)
if os.path.isdir(os.path.join(directory_path_sub_block, f)) and f not in
globals()[sub_block_folder_variable_name]]

        if sub_block_folders:
            if sub_block_folder_use_count == 0:
                selected_sub_block_folder = random.choice(sub_block_folders)
            globals()[sub_block_folder_variable_name].add(selected_sub_block_f
older)
            sub_block_folder_use_count += 1
            if sub_block_folder_use_count >= 5: # Need to be changed to 5 or
10 for the real experiment
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
                sub_block_folder_use_count = 0  # Reset the count for the next
folder
            sub_block_folder_path = os.path.join(directory_path_sub_block,
selected_sub_block_folder)
            image_files = [f for f in os.listdir(sub_block_folder_path) if
f.endswith((".jpg", ".png"))]
            image_files = sorted(image_files)
```

```
            # Extract variables from the first image file's name
            first_image_filename = image_files[0]
            file_info_1 =
extract_variables_from_filename_1(first_image_filename)

            for idx, image_file in enumerate(image_files):
                image_path = os.path.join(sub_block_folder_path, image_file)
                display_image(image_path)

                if idx == 0:
                    space_pressed = False
                    while not space_pressed:
                        for event in pygame.event.get():
                            if event.type == pygame.QUIT:
                                pygame.quit()
                                exit()
                            elif event.type == pygame.KEYDOWN and event.key ==
pygame.K_SPACE:
                                space_pressed = True
            STATE = blocks[block_index]
        else:
            if last_round < 4:
                last_round += 1
                sub_block_folder_path = os.path.join(directory_path_sub_block,
selected_sub_block_folder)
                image_files = [f for f in os.listdir(sub_block_folder_path) if
f.endswith((".jpg", ".png"))]
                image_files = sorted(image_files)

                # Extract variables from the first image file's name
                first_image_filename = image_files[0]
                file_info_1 =
extract_variables_from_filename_1(first_image_filename)

                for idx, image_file in enumerate(image_files):
                    image_path = os.path.join(sub_block_folder_path,
image_file)
                    display_image(image_path)

                    if idx == 0:
                        space_pressed = False
                        while not space_pressed:
                            for event in pygame.event.get():
                                if event.type == pygame.QUIT:
                                    pygame.quit()
                                    exit()
                                elif event.type == pygame.KEYDOWN and
event.key == pygame.K_SPACE:
                                    space_pressed = True
```

```
                    STATE = blocks[block_index]

            elif block_index < 5:
                block_durations = time.time() - start_time_block
                block_durations_min = block_durations / 60
                csv_row_T = [participant_number,
                    block_index,
                    file_info_1["Dif_Level"],
                    block_durations,
                    block_durations_min]
                csv_writer_T.writerow(csv_row_T)
                start_time_block = 0
                last_round = 0
                block_index += 1  # Move to the next block
                STATE = "Pause"
                start_time = time.time()  # Record the start time of the pause
            else:
                block_durations = time.time() - start_time_block
                block_durations_min = block_durations / 60
                csv_row_T = [participant_number,
                    block_index,
                    file_info_1["Dif_Level"],
                    block_durations,
                    block_durations_min]
                csv_writer_T.writerow(csv_row_T)
                start_time_block = 0
                start_time = time.time()
                STATE = "End"


    if STATE in blocks:
        # Use the current block_index to select the directory
        directory_path = shuffled_directory_paths[block_index]
        folder_variable_name = f"used_folders_R_{block_index}"
        Q1_variable_name = f"Q{block_index}yes"
        Q2_variable_name = f"Q{block_index}no"

        if not globals()[folder_variable_name]:
            globals()[folder_variable_name] = set()

        folders = [f for f in os.listdir(directory_path) if
os.path.isdir(os.path.join(directory_path, f)) and f not in
globals()[folder_variable_name]]

        if folders:
            selected_folder = random.choice(folders)
            globals()[folder_variable_name].add(selected_folder)
            folder_path = os.path.join(directory_path, selected_folder)
            image_files = [f for f in os.listdir(folder_path) if
f.endswith((".jpg", ".png"))]
```

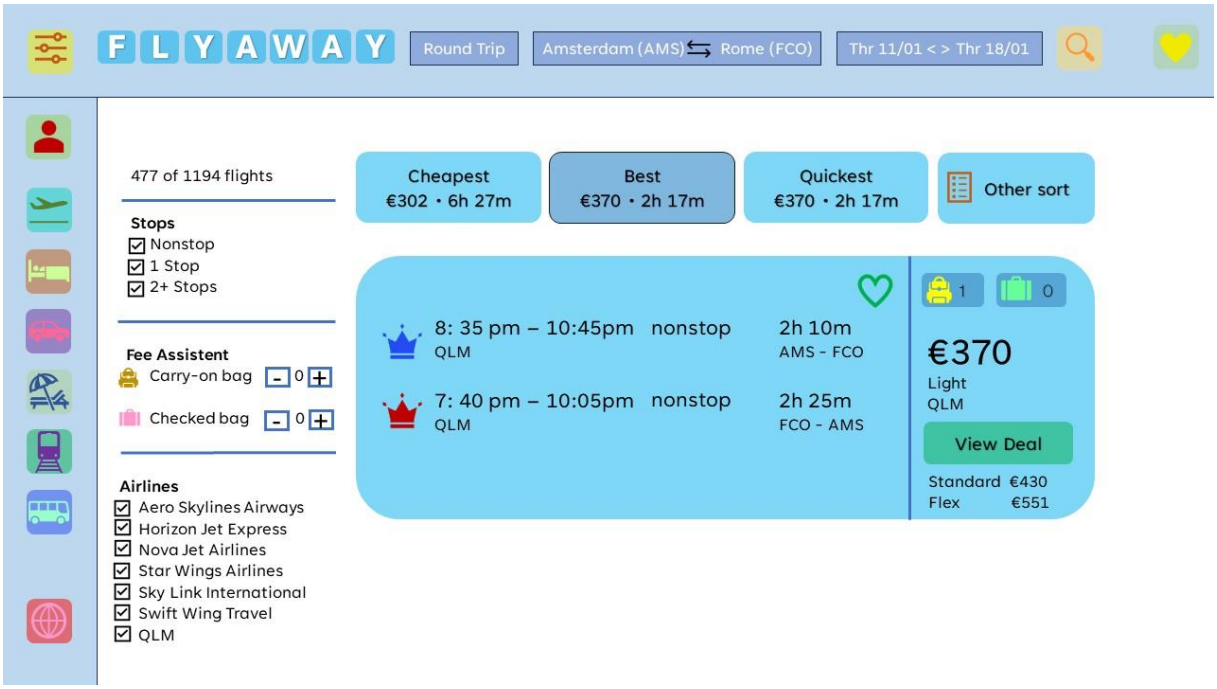
```
            image_files = sorted(image_files)

            # Extract variables from the first image file's name
            second_image_filename = image_files[0]
            file_info_2 =
extract_variables_from_filename_2(second_image_filename)

            for idx, image_file in enumerate(image_files):
                image_path = os.path.join(folder_path, image_file)
                display_image(image_path)

                if idx == 0:
                    if len(image_files) > 1:
                            pygame.time.delay(3000)
                            display_gray_screen(1000)
                else:
                    if idx == 1:
                        if len(image_files) > 1:
                            pygame.time.delay(3000)

            STATE = f"R_{block_index}_Question_1"

    if STATE == "Pause":
        current_time = time.time()
        if current_time - start_time < pause_duration:
            screen.fill(WHITE)
            draw_text("Time for a break", LARGE_FONT, BLACK, WIDTH // 2,
HEIGHT // 2)
            # Calculate and display the remaining time
            remaining_time = int(pause_duration - (current_time - start_time))
            draw_text(f"Time remaining: {remaining_time} seconds",MEDIUM_FONT,
BLACK, WIDTH // 2, HEIGHT // 2 + 50)
            draw_text(f"The experiment will automatically begin after the
break.",MEDIUM_FONT, BLACK, WIDTH // 2, HEIGHT // 2 + 100)
            sub_block_folder_use_count = 0
            pygame.display.flip()
            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN and event.key == pygame.K_q:
                    STATE = "Start_T"
        else:
            STATE = "Start_T"  # Return to the next block when the pause is
over


    if STATE in ["R_0_Question_1", "R_1_Question_1", "R_2_Question_1",
"R_3_Question_1", "R_4_Question_1", "R_5_Question_1"]:
        if not start_time_Q1:
            start_time_Q1 = time.time()
        block_index = int(STATE.split("_")[1])
        Q1_variable_name = f"Q{block_index}yes"
```

```
        Q2_variable_name = f"Q{block_index}no"
        # Handle button clicks
        x, y = pygame.mouse.get_pos()
        if button_yes.collidepoint(x, y):
            if pygame.mouse.get_pressed()[0]:
                Q1yes = 1
                Q1no = 0
                globals()[Q1_variable_name] = 1
                globals()[Q2_variable_name] = 0
        elif button_no.collidepoint(x, y):
            if pygame.mouse.get_pressed()[0]:
                Q1yes = 0
                Q1no = 1
                globals()[Q1_variable_name] = 0
                globals()[Q2_variable_name] = 1
        elif next_button.collidepoint(x, y) and (Q1yes or Q1no):
            if next_button.collidepoint(x, y) and (Q1yes or Q1no):
                if pygame.mouse.get_pressed()[0]:
                    time_taken_Q1 = time.time() - start_time_Q1
                    # Reset the variables
                    start_time_Q1 = 0
                    globals()[Q1_variable_name] = 0
                    globals()[Q2_variable_name] = 0
                    STATE = f"R_{block_index}_Question_2"
        screen.fill(GRAY)
        draw_buttons_1Q()
        draw_next_button_1Q()
        pygame.display.flip()

    if STATE in ["R_0_Question_2", "R_1_Question_2", "R_2_Question_2",
"R_3_Question_2", "R_4_Question_2", "R_5_Question_2"]:
        block_index = int(STATE.split("_")[1])
        Q1_variable_name = f"Q{block_index}yes"
        Q2_variable_name = f"Q{block_index}no"
        # Handle button clicks
        x, y = pygame.mouse.get_pos()
        if button_yes.collidepoint(x, y):
            if pygame.mouse.get_pressed()[0]:
                globals()[Q1_variable_name] = 1
                globals()[Q2_variable_name] = 0
                Q2yes = 1
                Q2no = 0
        elif button_no.collidepoint(x, y):
            if pygame.mouse.get_pressed()[0]:
                globals()[Q1_variable_name] = 0
                globals()[Q2_variable_name] = 1
                Q2yes = 0
                Q2no = 1
        elif next_button.collidepoint(x, y) and (Q2yes or Q2no):
            if pygame.mouse.get_pressed()[0]:
```

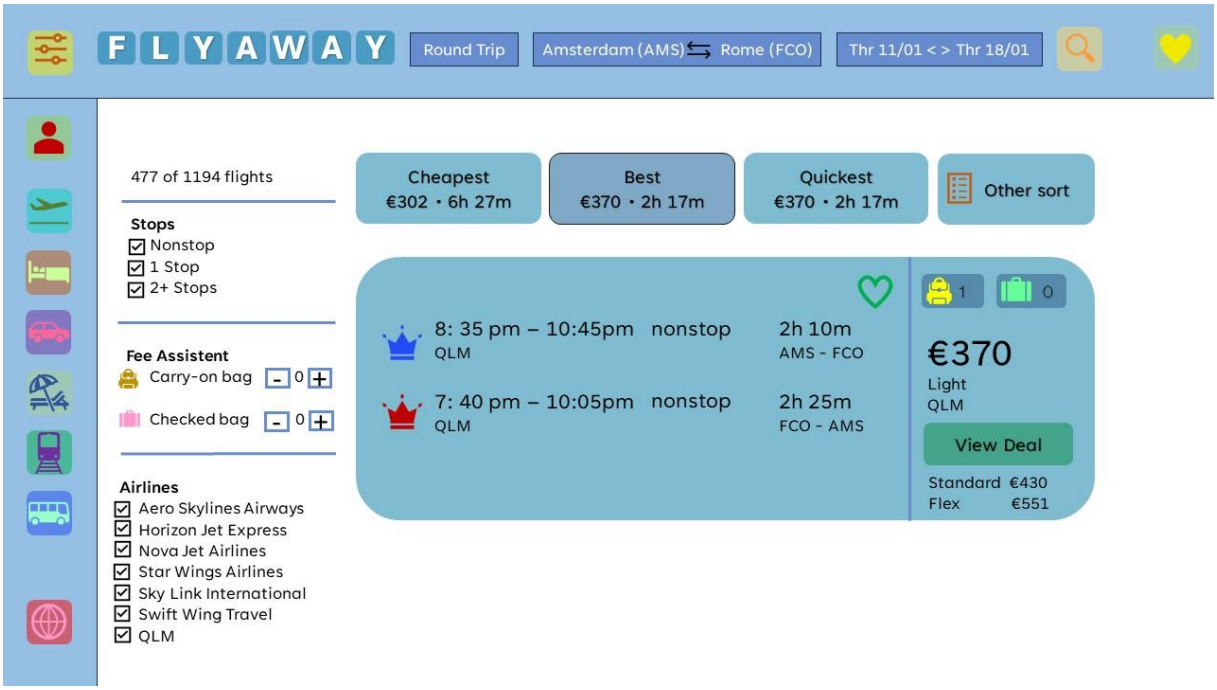
```
                # Calculate validation variables
                Q1_validate = int(Q1yes == file_info_1["Requirments"])
                Q2_validate = int(Q2yes == file_info_2["Change"])

                # Write participant data to the CSV file
                csv_row = [
                    participant_number,
                    block_index,
                    first_image_filename,
                    second_image_filename,
                    file_info_1["Requirments"],
                    file_info_2["Change"],
                    file_info_2["Fond"],
                    file_info_2["Colour"],
                    time_taken_Q1,
                    Q1yes,
                    Q2yes,
                    Q1_validate,
                    Q2_validate
                ]
                csv_writer.writerow(csv_row)

                # Reset the variables
                globals()[Q1_variable_name] = 0
                globals()[Q2_variable_name] = 0
                Q1yes = 0
                Q1no = 0
                time_taken_Q1 = 0
                Q2yes = 0
                Q2no = 0
                STATE = f"Sub_Block_{block_index}"
        screen.fill(GRAY)
        draw_buttons_2Q()
        draw_next_button_2Q()
        pygame.display.flip()

    if STATE == "End":
        start_time = time.time()
        if current_time - start_time < end_duration:
            draw_end()
        else:
            STATE = "Quit"

    if STATE == "Quit":
            pygame.quit()

pygame.quit()
```

Appendix B

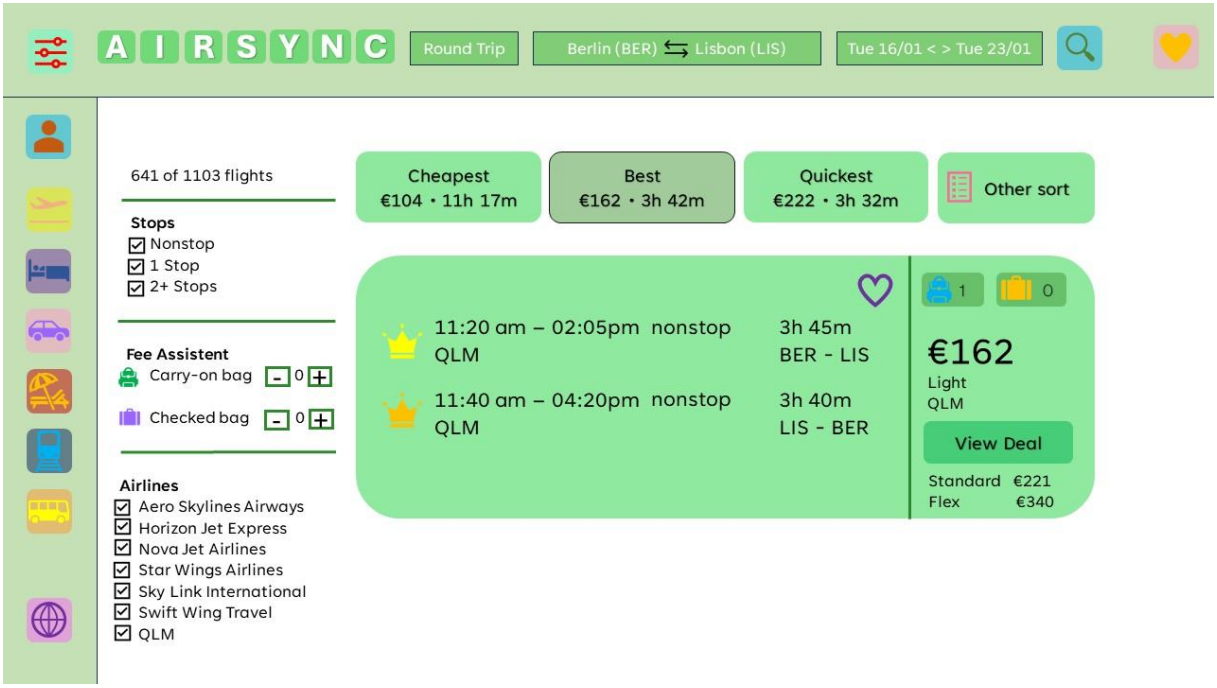
Colour changes



No change



Colour change

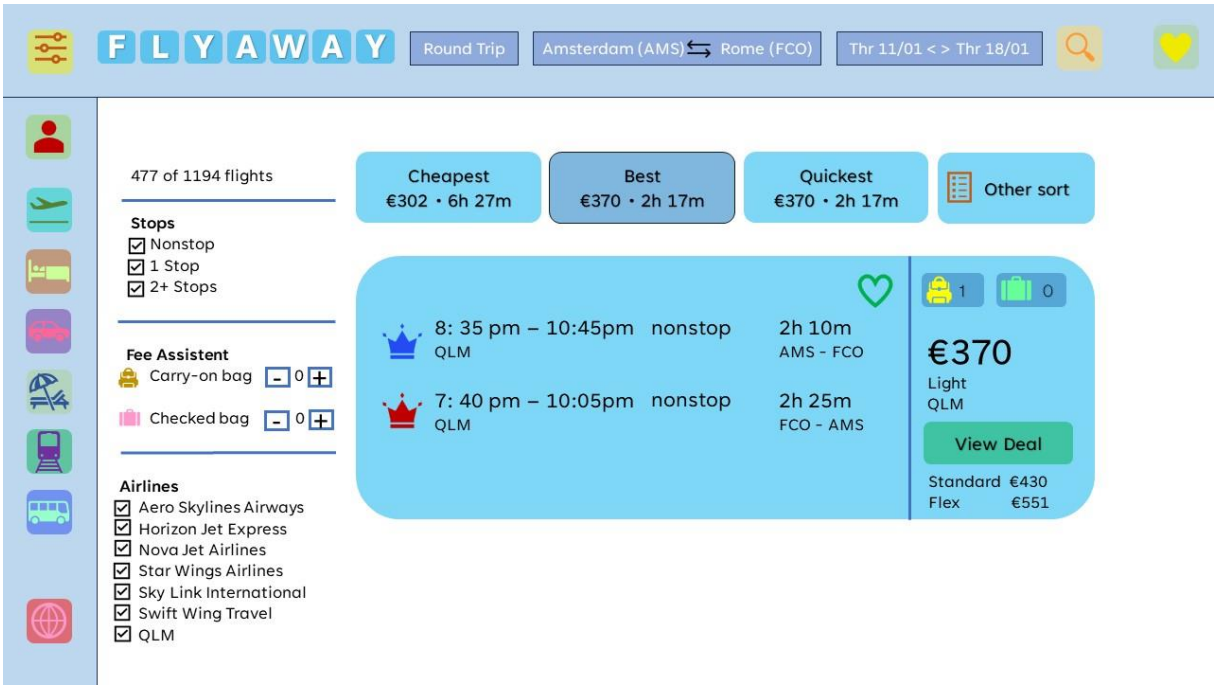


No change

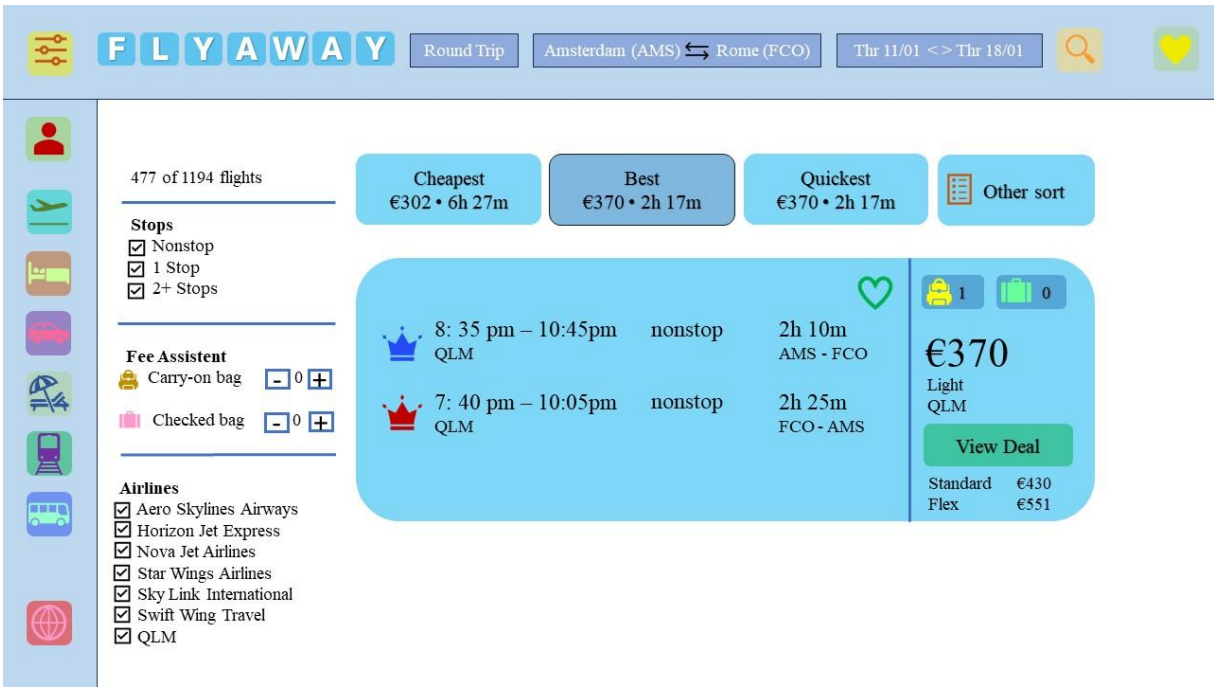
Colour change

Appendix C

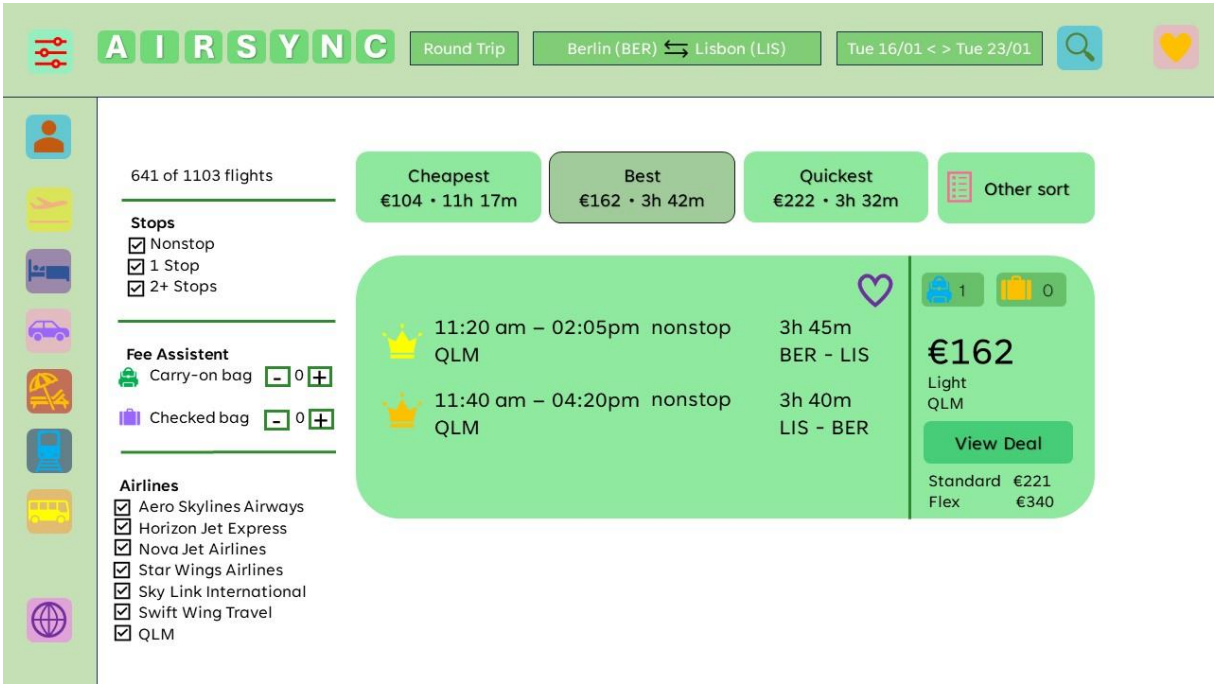
Font changes



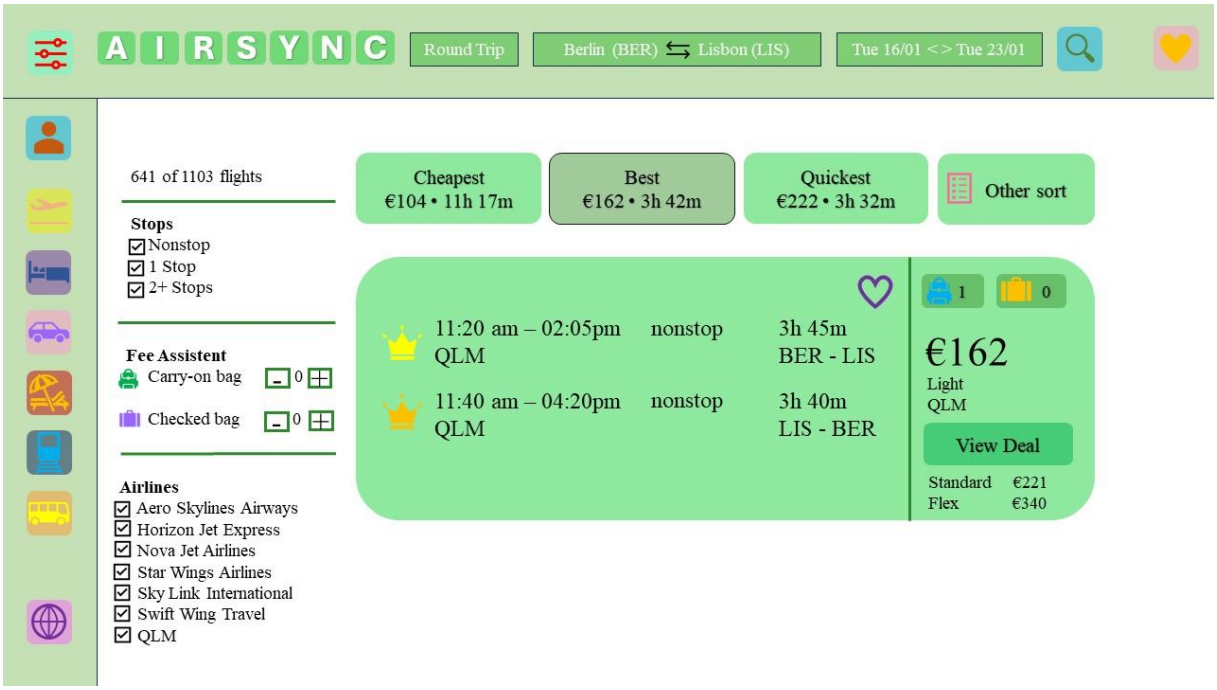
No change



Font change



No change



Font change

Font changes:

- Fond change 1: Tenorite - Times New Roman
- Fond change 2: Tenorite - Centaur
- Fond change 3: Tenorite - Congenial
- Fond change 4: Tenorite - Euphemia
- Fond change 5: Tenorite - Gill Sans MT

- Fond change 6: Tenorite - MingLiU
- Fond change 7: Tenorite - Segoe UI Emoji
- Fond change 8: Tenorite - Arial Narrow

Appendix D

Informed consent form

Welcome to my study!

You are invited to participate in the study “Exploring the Influence of Different Complexity Levels of Instructions on Visual Working Memory in Website Interaction”. This study is being done by Yannis Bittner from the bachelor’s program psychology, faculty Behavioural, Management, and Social Studies at the University of Twente. This study is supervised by Dr. Rob van der Lubbe.

Goal of the Study

The goal of this research study is to find out if different specified instructions have an impact on our ability to detect change on websites.

The Experiment

Your participation in this study involves a change blindness experiment, which will take approximately 40-60 minutes to complete.

Potential Risks and Discomforts

Participating in this study carries no physical, legal, or financial risks. The research has received approval from the Ethics Committee/domain Humanities & Social Sciences within the Faculty of Behavioural, Management, and Social Sciences.

Compensation

Compensation, in the form of SONA points, will be provided if you participate through SONA. Otherwise, there is no compensation for taking part in this study.

Confidentiality

Your privacy will be safeguarded to the best of my ability. Confidential or personal data will not be disclosed in a manner that could identify you. Data collected will be anonymized or assigned pseudonyms in any publication. All research data will be securely stored at the University of Twente on encrypted devices for a maximum of 3 years. After this period, data will either be deleted or anonymized to prevent identification. If necessary, and for purposes of scientific integrity checks, research data may be made available in anonymous form to external parties.

Voluntary Participation and Withdrawal

Your participation in this study is entirely voluntary. You have the right to withdraw at any time or decline the use of your data for the study, without the need for an explanation. Withdrawal will not have any negative consequences.

Contact Information

If you have any questions, concerns, or wish to withdraw from the study, please contact:

Yannis Bittner at y.l.bittner@student.utwente.nl

For questions about your rights as a participant or other study-related matters, you may contact the Ethics Committee/domain Humanities & Social Sciences for the faculty of Behavioural, Management, and Social Sciences at the University of Twente at ethicscommittee-hss@utwente.nl.

For inquiries regarding the confidentiality of your personal information, you may also reach out to the Data Protection Officer at the University of Twente at dpo@utwente.nl.

By giving your consent, you agree to all the following statements:

• I have read and understood the information for this study, or it has been read to me. Any and all questions I had about this study have been asked and answered to my satisfaction.

• I voluntarily partake in this study. There is no explicit or implicit pressure for me to partake in this study. It is clear to me that I can withdraw my participation at any time without giving a reason. I do not have to answer any question I wish not to answer.

• I understand that the data I provide during this research will be analysed for research into the topic of visual working memory on university websites.

• I understand that personal information that can identify me (i.e., my name or personal details) will not be shared beyond this research team.

• I am 18 years or older.

Appendix E

Instructions for all difficulty levels and both website types

	Difficulty level					
	1		2		3	
	Match	No-match	Match	No-match	Match	No-match
AMS	- You are looking for a flight from Amsterdam to Rome. - You are looking for a vacation between 11.01 and 18.01. - You are looking for a holiday with a flight budget of 400€.	- Your flight budget is 250€. - Your budget is 430€ but you want a Flex ticket. - You are looking for a flight leaving Amsterdam before 1:10 p.m.	- You are looking for a flight from Amsterdam to Rome. On the way back you want to be in Amsterdam around 10 p.m. - You are looking for a flight from Amsterdam to Rome, leaving Amsterdam around 8:30 p.m. - You want to go on a holiday for a maximum of 390€ with an included carry-on bag for that price.	- You are looking for a flight from the 16.02 to the 23.02. Your flight budget is 270€. - You are looking for the cheapest flight from Amsterdam to Ljubljana from the 27.02 to the 04.03. - You are looking for a flight between 16.03 and 23.03. Your budget is 450€ but you want a Flex ticket.	- You are looking for a flight from Amsterdam to Rome from 11.01 to 18.01. Your flight budget is 400€. - You are looking for a flight from Amsterdam to Rome between 11.01 and 18.01. On the way back you want to be in Amsterdam by 10 p.m. - You are looking for a flight from Amsterdam to Rome, leaving Amsterdam after 07 p.m. and you don't want to pay more than 380€.	- You are looking for a flight from Amsterdam to Lisbon from 15.01 to 23.0. Your flight budget is 300€. - You are looking for a flight from London to Rome leaving Rome before 4 p.m. and operated by Star Wings Airlines. - You want to fly from Amsterdam to Athens for a Maximum of 320€ with an included checked bag for that price.
BER	- You are looking for a flight from Berlin to Lisbon. - You are looking for a vacation between 16.01 to 23.01. - You are looking for a holiday with a flight	- Your flight budget is 100€. - Your budget is 320€ but you want a Flex ticket. - You are looking for a flight leaving Berlin at 2 p.m.	- You are looking for a flight from Berlin to Lisbon from the 16.01 to the 23.01. - You are looking for a flight from Berlin to Lisbon. On the way back you want to be in Berlin before 5 p.m.	- You are looking for a flight from Paris to Lisbon leaving Lisbon after 8 p.m. - You want to fly for a maximum of 150€ with an included checked bag for that price. - You are looking for a flight for 100€	- You are looking for a flight from Berlin to Lisbon from the 16.01 to the 23.01. Your flight budget is 170€. - You are looking for a flight from Berlin to Lisbon between 16.01 and 23.01. On the way back you want to be in Berlin before 5 p.m. - You are looking for a flight from Berlin to Lisbon, Leaving Berlin after 10 a.m. and you don't want to pay more than 180€.	- You are looking for a flight from Berlin to Dublin, leaving Berlin at 3 p.m. and you don't want to pay more than 100€. - You are looking for the cheapest flight from Berlin to Oslo from the 27.12 to the 04.01. - You are looking for a flight from Paris to Lisbon leaving Lisbon at 4 p.m. You just want to fly Horizon Jet Express.

	budget of 180€.		- You are looking for a flight from Berlin to Lisbon, Leaving Berlin after 10 a.m.	including 2 carry-on bags.		

Appendix F

All block-type orders.

Order number	Order
01	"1_AMS", "2_BER", "3_AMS", "1_AMS", "2_BER", "3_AMS"
02	"1_BER", "2_AMS", "3_BER", "1_BER", "2_AMS", "3_BER"
03	"1_AMS", "3_BER", "2_AMS", "1_AMS", "3_BER", "2_AMS"
04	"1_BER", "3_AMS", "2_BER", "1_BER", "3_AMS", "2_BER"
05	"2_AMS", "1_BER", "3_AMS", "2_AMS", "1_BER", "3_AMS"
06	"2_BER", "1_AMS", "3_BER", "2_BER", "1_AMS", "3_BER"
07	"2_AMS", "3_BER", "1_AMS", "2_AMS", "3_BER", "1_AMS"
08	"2_BER", "3_AMS", "1_BER", "2_BER", "3_AMS", "1_BER"
09	"3_AMS", "1_BER", "2_AMS", "3_AMS", "1_BER", "2_AMS"
10	"3_BER", "1_AMS", "2_BER", "3_BER", "1_AMS", "2_BER"
11	"3_AMS", "2_BER", "1_AMS", "3_AMS", "2_BER", "1_AMS"
12	"3_BER", "2_AMS", "1_BER", "3_BER", "2_AMS", "1_BER"

Appendix G

R Code

```
library(writexl)
library(readxl)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(tibble)
library(psycho)
library(rstatix)
library(plotrix)
library(Rmisc)

# Survey data analysis
APSD <- read_csv("Bachelor Thesis Yannis Bittner_November 27, 2023_11.54.csv")
#APSD <- read_csv("Default Report.csv" , sep = ",")

# Selecting variables
APSD = APSD %>% filter(Finished == 1)
APSD = APSD %>% dplyr::select(18:28, -19, -26)

# Oreperation
APSD$`Participant number` = as.factor(APSD$`Participant number`)
APSD$Age = as.numeric(APSD$Age)
APSD$Colour_blindness_test = as.numeric(APSD$Colour_blindness_test)
APSD$Nationality = as.numeric(APSD$Nationality)
APSD$Nationality_Text = as.factor(APSD$Nationality_Text)
APSD$Highest_degree = as.numeric(APSD$Highest_degree)
APSD$Sex = as.numeric(APSD$Sex)
APSD$Gender_identity = as.numeric(APSD$Gender_identity)
```

```
APSD$Vision_test = as.factor(APSD$Vision_test)

APSD_24 = APSD %>% slice(-c(1,2,3))

# Analysing survey data
APSD %>%
  summary()

APSD %>% summarise(mean = mean(Age), sd = sd(Age),
                   var = var(Age), minimum = min(Age), maximum = max(Age))

APSD %>% summarise(mean = mean(Nationality), sd = sd(Nationality),
                   var = var(Nationality), minimum = min(Nationality), maximum =
max(Nationality))
APSD %>% summarise(mean = mean(Sex), sd = sd(Sex),
                   var = var(Sex), minimum = min(Sex), maximum = max(Sex))
APSD %>% summarise(Sex)

APSD_24$Sex %>% table()
APSD$Sex %>% prop.table()

# Experiment data analysis
# Empty list to store individual participant data frames
participant_data = list()

# Loop through participant files and read them into individual data frames
for (i in 4:27) {
  file_name = paste0("experiment_data_participant_", i, ".csv")
  participant_data[[i]] = read.csv(file_name, header = TRUE)
}
```

```
# Combine all participant data frames into a single data frame
c_data = do.call(rbind, participant_data)

# Creating varible Dif_level and Conditions
c_data$Dif_level = as.integer(gsub("[^0-9]", "", c_data$First_Picture_Filename))
c_data = mutate(c_data, AMS = ifelse(grepl("A", First_Picture_Filename), 1, 0))
c_data$Conditions = ifelse(c_data$Change == 1 & c_data$Colour == 1, "Colour",
                    ifelse(c_data$Change == 1 & c_data$Colour == 0, "Font", NA))

# Calculating overall false alarms by participant and by participant and block
o_Dif_false_alarms = c_data %>%
  group_by(Participant_Number, Dif_level) %>%
  summarise(
    false_alarms = sum(Change == 0 & Q2 == 1)
  )

o_Blo_false_alarms = c_data %>%
  group_by(Participant_Number, Dif_level, Block_Number) %>%
  summarise(
    false_alarms = sum(Change == 0 & Q2 == 1)
  )

# Calculating overall correct rejections by participant and by participant and block
o_Dif_correct_rejections <- c_data %>%
  group_by(Participant_Number, Dif_level) %>%
  summarise(
    correct_rejections = sum(Change == 0 & Q2 == 0)
  )

o_Blo_correct_rejections <- c_data %>%
  group_by(Participant_Number, Dif_level, Block_Number) %>%
```

```
  summarise(
    correct_rejections = sum(Change == 0 & Q2 == 0)
  )


# Combining data sets
nt_c_data = c_data %>% left_join(o_Dif_correct_rejections %>%
dplyr::select(Participant_Number, Dif_level, correct_rejections), by =
c("Participant_Number", "Dif_level"))
nt_c_data = nt_c_data %>% left_join(o_Dif_false_alarms %>%
dplyr::select(Participant_Number, Dif_level, false_alarms), by = c("Participant_Number",
"Dif_level"))
bnt_c_data = c_data %>% left_join(o_Blo_correct_rejections %>%
dplyr::select(Participant_Number, Dif_level, Block_Number, correct_rejections), by =
c("Participant_Number", "Dif_level", "Block_Number"))
bnt_c_data = bnt_c_data %>% left_join(o_Blo_false_alarms %>%
dplyr::select(Participant_Number, Dif_level, Block_Number, false_alarms), by =
c("Participant_Number", "Dif_level", "Block_Number"))


# Data set for calculation of d Prime by participants
result_6 = nt_c_data %>%
  group_by(Participant_Number, Dif_level, AMS, Conditions) %>%
  reframe(
    hits = sum(Change == 1 & Q2 == 1),
    misses = sum(Change == 1 & Q2 == 0),
    false_alarms = false_alarms,
    correct_rejections = correct_rejections
  ) %>% distinct() %>% filter(!is.na(Conditions))


# Data set for calculation of d Prime by participants and blocks
result_7 = bnt_c_data %>%
  group_by(Participant_Number, Block_Number, Block_Set, Dif_level, AMS, Conditions)
%>%
  reframe(
    hits = sum(Change == 1 & Q2 == 1),
```

```
    misses = sum(Change == 1 & Q2 == 0),
    false_alarms = false_alarms,
    correct_rejections = correct_rejections
  ) %>% distinct() %>% filter(!is.na(Conditions))

# Calculating for d prime
# Specify the file path where you want to save the Excel file
file_path2 <- "G:/My Drive/Bachelor PSY year 4/Bachelor Thesis/Python/Experiment/Participant data/result_6.xlsx"
file_path3 <- "G:/My Drive/Bachelor PSY year 4/Bachelor Thesis/Python/Experiment/Participant data/result_7.xlsx"

# Export 'result_dprime_2' dataset to Excel
#write_xlsx(result_6, file_path2)
#write_xlsx(result_7, file_path3)

# Load the Excel file into a data frame
data_2 <- read_excel(file_path2)
data_3 <- read_excel(file_path3)
data_2$AMS = ifelse(data_2$AMS == 1, "AMS", "BER")
data_3$AMS = ifelse(data_3$AMS == 1, "AMS", "BER")

# Shorten the data sets
new_s_d_prime_data = data_2 %>% dplyr::select(1:4, 13)
nb_d_prime_data = data_3 %>% dplyr::select(1:6, 15)
beta = data_2 %>% dplyr::select(1:4, 14)
beta_LE = data_3 %>% dplyr::select(1, 3, 4, 6, 16)

# Analysis
# Preperation
new_s_d_prime_data$Participant_Number = as.factor(new_s_d_prime_data$Participant_Number)
```

```
new_s_d_prime_data$Dif_level = as.factor(new_s_d_prime_data$Dif_level)
new_s_d_prime_data$AMS = as.factor(new_s_d_prime_data$AMS)
new_s_d_prime_data$Conditions = as.factor(new_s_d_prime_data$Conditions)

beta$AMS = as.factor(beta$AMS)
beta$Conditions = as.factor(beta$Conditions)

# Normality assumption
new_s_d_prime_data %>% shapiro_test(d_prime)

# Homogeneity of variance assumption
new_s_d_prime_data %>%
  levene_test(d_prime ~ AMS)
new_s_d_prime_data %>%
  levene_test(d_prime ~ Conditions)
new_s_d_prime_data %>%
  levene_test(d_prime ~ Dif_level)

#t-test
t.test(d_prime ~ AMS, data = new_s_d_prime_data)

# Within ANOVA analysis
result.aov_01 = new_s_d_prime_data %>%
  anova_test(
    dv = d_prime,
    wid = Participant_Number,
    within = c(Dif_level, Conditions)
  )

get_anova_table(result.aov_01)
```

```
# Within ANOVA analysis for beta
result.aov_03 = beta %>%
  anova_test(
    dv = Beta,
    wid = Participant_Number,
    within = c(Dif_level, Conditions)
  )

get_anova_table(result.aov_03)

# Within ANOVA for learning effect
result.aov_02 = nb_d_prime_data %>%
  anova_test(
    dv = d_prime,
    wid = Participant_Number,
    within = c(Dif_level, Conditions , Block_Set)
  )

get_anova_table(result.aov_02)

# Within ANOVA for learning effect with beta
result.aov_04 = beta_LE %>%
  anova_test(
    dv = Beta,
    wid = Participant_Number,
    within = c(Dif_level, Conditions , Block_Set)
  )

get_anova_table(result.aov_04)

# Graphics
```

```
# Bar plot preparation

d_Dif_Con = summarySE(new_s_d_prime_data, measurevar="d_prime",
groupvars=c("Dif_level", "Conditions"))

d_Blo_Con = summarySE(nb_d_prime_data, measurevar="d_prime",
groupvars=c("Block_Set", "Conditions"))

b_Dif_Con = summarySE(beta, measurevar="Beta", groupvars=c("Dif_level", "Conditions"))

b_Blo_Con = summarySE(beta_LE, measurevar="Beta", groupvars=c("Block_Set",
"Conditions"))


# Bar plot with error bars d prime with conditions and difficulty levels

ggplot(d_Dif_Con, aes(x=Conditions, y=d_prime, fill=Dif_level)) +
  geom_bar(position=position_dodge(), stat="identity", colour="black", linewidth=.3) +
  geom_errorbar(aes(ymin=d_prime-se, ymax=d_prime+se),
                width=.2,
                position=position_dodge(.9)) +
  scale_y_continuous(limits=c(0, 2.5)) +
  scale_fill_hue(name="Difficulty Level") +
  labs(y="d Prime",
       x="Change" ) +
  theme_classic()


# Bar plot with error bars beta with conditions and difficulty levels

ggplot(b_Dif_Con, aes(x=Conditions, y=Beta, fill=Dif_level)) +
  geom_bar(position=position_dodge(), stat="identity", colour="black", linewidth=.3) +
  geom_errorbar(aes(ymin=Beta-se, ymax=Beta+se),
                width=.2,
                position=position_dodge(.9)) +
  scale_y_continuous(limits=c(0, 2.5)) +
  scale_fill_hue(name="Difficulty Level") +
  labs(y="Beta",
       x="Change" ) +
  theme_classic()
```

```
# Bar plot with error bars d prime with conditions and block set
ggplot(d_Blo_Con, aes(x=Conditions, y=d_prime, fill=Block_Set)) +
  geom_bar(position=position_dodge(), stat="identity", colour="black", linewidth=.3) +
  geom_errorbar(aes(ymin=d_prime-se, ymax=d_prime+se),
                width=.2,
                position=position_dodge(.9)) +
  scale_y_continuous(limits=c(0, 2.5)) +
    scale_fill_hue(name="Try") +
  labs(y="d Prime",
       x="Change" ) +
  theme_classic()

# Bar plot with error bars beta with conditions and block set
ggplot(b_Blo_Con, aes(x=Conditions, y=Beta, fill=Block_Set)) +
  geom_bar(position=position_dodge(), stat="identity", colour="black", linewidth=.3) +
  geom_errorbar(aes(ymin=Beta-se, ymax=Beta+se),
                width=.2,
                position=position_dodge(.9)) +
  scale_y_continuous(limits=c(0, 2.5)) +
  scale_fill_hue(name="Try") +
  labs(y="Beta",
       x="Change" ) +
  theme_classic()
```